Robot Visual Servo with Fuzzy Particle Filter

Jie Ma and Qingjie Zhao
Beijing Lab of Intelligent Information Technology
School of Computer Science, Beijing Institute of Technology, Beijing, China

Abstract—**This paper proposes a robot visual servo method with an adaptive particle filter based on fuzzy logic theory to online estimate the total Jacobian matrix of a robot visual servo system. A set of fuzzy rules are used to select appropriate numbers of particles according to the filtering estimation errors. When an estimation error is high more particles are used, and when the estimation error is low fewer particles are used. The visual servo results on a two degree-of-freedom robot system show that the proposed fuzzy adaptive particle filter visual servo method needs less time than that of traditional particle filter visual servo method to get a comparative tracking accuracy.**

Index Terms—**visual servo, particle filter, fuzzy logic**

I. Introduction

Image-based robot visual servo control method [1-3] does not require three-dimensional reconstruction of an object, and the control error is directly defined from the difference between current image features and expected image features. The control strategy usually requires an image Jacobian matrix known. Traditional image Jacobian matrix relates changes in joint angles (or robot's positions) to changes in image features. For an eye-in-hand structure, movements of the robot end-effector will always lead to changes of object's image features. When an object is moving, both movements of the end-effector and the object can cause changes in image features [4]. Therefore total Jacobian matrix in [4] is defined to include the factor of object motion.

There are several online estimation techniques to acquire image or total Jacobian matrix. For static or slow objects, Hosoda [5] uses exponentially weighted recursive least square update method to estimate the Jacobian matrix; Jagersand [6] takes a trust region control scheme in which the Jacobian is updated using Broyden's method. When tracking a moving target, Piepmeier [7] demonstrates a dynamic quasi-Newton method to control a robot tracking a moving target, where a stationary camera is used in the workspace. For an eye-in-hand case, observed image feature changes of an object may result from camera (or manipulator) motion, or from target motion, or both of them. It is difficult to decide how many changes are caused by target moving if no additive restrictions used. Asada [8] utilizes two cameras on the robot effecter to track an unknown moving object, which needs three additional stationary marks to predict the motion of the virtually stationary target. Zhao [4] proposes total Jacobian consisting of image Jacobian and object Jacobian and estimates the total Jacobian matrix using exponentially weighted least square algorithm.

Especially, Qian [9] and Lv [10] establish a state space model where the state is composed of the elements of the image Jacobian matrix, and thus converts image Jacobian estimation into system state estimation by using Kalman filter, requiring that the system is linear and with Gaussian white noise. As an improvement, zhao [11-12] proposes a particle filter estimation method to get the total Jacobian matrix for general non-linear and non-Gaussian systems.

In this paper, fuzzy logic theory is introduced into particle filter framework. A fuzzy adaptive particle filter is used to estimate image Jacobian matrix and a novel un-calibrated robot visual servo method is proposed.

II. State space model

Suppose $\boldsymbol{y}_d \in \Re^m$ is a desired image feature vector. For a moving object, image feature vector $\boldsymbol{y}$ is a function of both robot joint vector $\boldsymbol{\theta} = [\theta_1, \theta_2, \cdots, \theta_n]^T$ and time t, that is

$$\boldsymbol{y} = \boldsymbol{y}(\boldsymbol{\theta}, t). \tag{1}$$

Expanding (1) about $(\boldsymbol{\theta}, t)$ with Taylor series and leaving out those higher order terms yields:

$$\Delta\boldsymbol{y} = \boldsymbol{J}_\theta(\boldsymbol{\theta}, t)\Delta\boldsymbol{\theta} + \boldsymbol{J}_t(\boldsymbol{\theta}, t)\Delta t, \tag{2}$$

where

$$\boldsymbol{J}_\theta = \partial\boldsymbol{y}/\partial\boldsymbol{\theta}, \ \boldsymbol{J}_t = \partial\boldsymbol{y}/\partial t,$$

$$\Delta\boldsymbol{y} = \boldsymbol{y}(k) - \boldsymbol{y}(k-1),$$

$$\Delta\boldsymbol{\theta} = \boldsymbol{\theta}(k) - \boldsymbol{\theta}(k-1).$$

Δt is time interval, $\boldsymbol{J}_\theta$ is the traditional image Jacobian, and $\boldsymbol{J}_t$ is called object Jacobian. In (2) the first term is caused by camera motion, and the second term is caused by object motion. Equation (2) can be rewritten as:

$$\Delta\boldsymbol{y} = \begin{bmatrix}\boldsymbol{J}_\theta & \boldsymbol{J}_t\end{bmatrix} \cdot \begin{bmatrix}\Delta\boldsymbol{\theta} \\ \Delta t\end{bmatrix}. \tag{3}$$

Manuscript received January 16, 2010; revised June 1, 2010; accepted July 1, 2010. This work was supported by the National Natural Science Foundation of China (60772063, 61175096). Corresponding author: Qingjie Zhao (e-mail: qingjie.zhao@gmail.com)

doi:10.4304/jcp.7.4.842-845

We define $\boldsymbol{J} = [\boldsymbol{J}_\theta, \boldsymbol{J}_t]$ as total Jacobian. Suppose

$$\Delta\widetilde{\boldsymbol{\theta}} = [\Delta\boldsymbol{\theta}, \Delta t]^T ,$$

then (3) becomes as:

$$\Delta\boldsymbol{y} = \boldsymbol{J} \cdot \Delta\widetilde{\boldsymbol{\theta}} , \quad (4)$$

$$\boldsymbol{J} = \begin{bmatrix} \frac{\partial y_1}{\partial \theta_1} \cdots \frac{\partial y_1}{\partial \theta_n} \frac{\partial y_1}{\partial t} \\ \cdots \\ \frac{\partial y_m}{\partial \theta_1} \cdots \frac{\partial y_m}{\partial \theta_n} \frac{\partial y_m}{\partial t} \end{bmatrix} = \begin{bmatrix} J_{11} \cdots J_{1(n+1)} \\ \cdots \\ J_{m1} \cdots J_{m(n+1)} \end{bmatrix} .$$

In order to estimate the elements of total Jacobian matrix J, state vector x is defined as:

$$\boldsymbol{x} = [J_{11}, \cdots, J_{1(n+1)}, \cdots, J_{m1}, \cdots, J_{m(n+1)}]^T . \quad (5)$$

This is a $m(n+1)\times 1$ vector. According to (4) the state space model is:

$$\boldsymbol{x}(t) = \boldsymbol{x}(t-1) + \boldsymbol{\omega}(t-1) \quad (6)$$

$$\boldsymbol{z}(t) = \boldsymbol{H}(t)\boldsymbol{x}(t) + \boldsymbol{\upsilon}(t) \quad (7)$$

$$\boldsymbol{H}(t) = \begin{bmatrix} \Delta\widetilde{\boldsymbol{\theta}}(t)^T & & 0 \\ & \cdots & \\ 0 & & \Delta\widetilde{\boldsymbol{\theta}}(t)^T \end{bmatrix}_{m\times m(n+1)}$$

where (6) is state transfer equation and (7) is measurement equation. $\boldsymbol{z}(t)$ is measurement vector, and $\boldsymbol{\omega}(t)$ and $\boldsymbol{\upsilon}(t)$ are process noise and measurement noise respectively.

If state x is acquired by using estimation method according to the measurement values of robot joint angles and image features, the Jacobian matrix J is acquired also.

III. FUZZY PARTICLE FILTER

A. Traditional Particle Filter

Particle filter is a novel filtering algorithm based on recursive Bayesian estimation [13], especially suitable for solving nonlinear and non-Gaussian estimation problems. The basic idea of particle filter is that sufficient random samples can be utilized to represent the probability distribution of a state. Based on the measurements, real probability distribution is approximately achieved by regulating samples' weights and positions.

N samples of $\boldsymbol{x}(t)$ are produced, and every sample is given a weight according to its importance. Thus N pairs of particles $\{\boldsymbol{x}_t^i, \boldsymbol{w}_t^i\}(i = 1, \cdots, N)$ are produced, where $\boldsymbol{w}_t^i$ is called the importance weight of the ith sample $\boldsymbol{x}_t^i$ at time t. At the beginning, that is t=0, N initial samples $\boldsymbol{x}_0^i$ (i=1, …, N) are produced from $p(\boldsymbol{x}_0)$, and suppose $\boldsymbol{w}_0^i = 1/N$. For traditional particle filter algorithm, at every time t, the following steps are executed:

Step1. N prediction values $\hat{\boldsymbol{x}}_t^i$ (i=1, …, N) of the state $\boldsymbol{x}(t)$ are produced from the state transfer equation, corresponding N weights $\hat{\boldsymbol{w}}_t^i$ (i=1, …, N) are given from the measurement equation, and the weights are normalized to 0~1, that is

$$\hat{\boldsymbol{w}}_t^i \leftarrow \hat{\boldsymbol{w}}_t^i / \sum_{i=1}^{N} \hat{\boldsymbol{w}}_t^i .$$

Here, we get N particles $\{\hat{\boldsymbol{x}}_t^i, \hat{\boldsymbol{w}}_t^i\}(i = 1, \cdots, N)$.

Step2. N new samples $\boldsymbol{x}_t^i (i = 1, \cdots, N)$ are generated from $\{\hat{\boldsymbol{x}}_t^i, \hat{\boldsymbol{w}}_t^i\}(i = 1, \cdots, N)$ by replicating those samples with high weights and throwing off those samples with low weights. Each new sample is allocated a new weight $\boldsymbol{w}_t^i$=1/N. Then we get N new particles

$$\{\boldsymbol{x}_t^i, \boldsymbol{w}_t^i\}(i = 1, \cdots, N) .$$

Step3. A state estimation result is obtained, that is,

$$\boldsymbol{x}_t = \sum_{i=1}^{N} \boldsymbol{w}_t^i \boldsymbol{x}_t^i .$$

B. Fuzzy particle filter

The accuracy of particle filter is related to the number of particles, which is generally selected by experience or according to actual requirements. Generally speaking, the more particles, the higher the accuracy obtained. However, more particles mean burdensome computation. Therefore, increasing the number of particles to acquire a good estimation is not an optimal idea. We try to find a tradeoff between the number of particles and the accuracy required. In this paper a fuzzy logic method is used to dynamically modify the number of particles according to the filtering estimation error.

We define $e = \|\Delta\boldsymbol{y}\| - \|\boldsymbol{z}\|$, and the number of particles can be adjusted by $N \leftarrow (1+ke)N$. A set of fuzzy rules is constituted to adaptively adjust the number of particles according to e:

Rule1: IF e is NL, then N is NL;

Rule2: IF e is NS, then N is NS;

Rule3: IF e is M, then N is M;

Rule4: IF e is PS, then N is PS;

Rule5: IF e is PL, then N is PL.

Above e is the input linguistic variable, and N is the output linguistic variable. Their value is {NL, NS, M, PS, PL}. NL means negative large, NS means negative small, M means medium, PS means positive small, and PL means positive large. In order to limit the number of particles increasing infinitely, we give a maximum number of particles. The membership function shape of e and N is defined as in Fig.1.

We include the fuzzy particle filter algorithm as follows:

1). N prediction values $\hat{\boldsymbol{x}}_t^i$ (i=1, …, N) of the state $\boldsymbol{x}(t)$ are produced from the state transfer equation, corresponding N weights $\hat{\boldsymbol{w}}_t^i$ (i=1, …, N) are given from the measurement equation, and the weights are normalized to 0~1, that is

$$\hat{\boldsymbol{w}}_t^i \leftarrow \hat{\boldsymbol{w}}_t^i / \sum_{i=1}^{N} \hat{\boldsymbol{w}}_t^i .$$

Here, we get N particles $\{\hat{\boldsymbol{x}}_t^i, \hat{\boldsymbol{w}}_t^i\}(i=1,\cdots,N)$.

2). N new samples $\boldsymbol{x}_t^i(i=1,\cdots,N)$ are generated from $\{\hat{\boldsymbol{x}}_t^i, \hat{\boldsymbol{w}}_t^i\}(i=1,\cdots,N)$ by replicating those samples with high weights and throwing off those samples with low weights. Each new sample is allocated a new weight $\boldsymbol{w}_t^i$=1/N. Then we get N new particles

$$\{\boldsymbol{x}_t^i, \boldsymbol{w}_t^i\}(i=1,\cdots,N).$$

3). A state estimation result is obtained, that is,

$$\boldsymbol{x}_t = \sum_{i=1}^{N} \boldsymbol{w}_t^i \boldsymbol{x}_t^i .$$

4). The measurement is acquired from Equation (7), and *e* is computed by using $e = \|\Delta \boldsymbol{y}\| - \|\boldsymbol{z}\|$;

5). A new number of particles is gained by using the fuzzy rules.

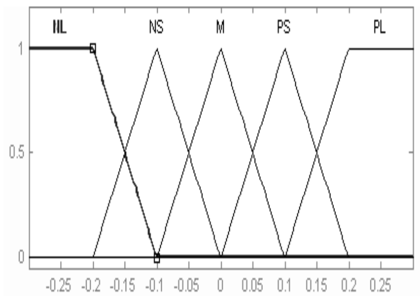

Figure 1. Membership function

IV. CONTROL METHOD

The task of visual servo is to control the motion of a robot via proper control law according to the measured image feature error, and eventually make the image feature error approximate to zero.

The image feature error is obtained through the desired image features subtracted by the current image features. The changes of image features and the robot joints are used to estimate the current state through the fuzzy adaptive particle filter and so the Jacobian matrix is obtained. The robot is controlled by using the inverse Jacobian matrix. The process is repeated until the image feature error approximates to zero. For a two degree-of-freedom system, the control process is:

1) Capturing a picture of the object, reading robot's joint angles $\boldsymbol{\theta} = [\theta_1, \theta_2]^T$ and computing $\Delta\boldsymbol{\theta} = \boldsymbol{\theta}(t) - \boldsymbol{\theta}(t-1)$;

2) Computing the image features of the object, such as its center coordinate $\boldsymbol{y} = [y_1, y_2]^T$;

3) Calculating $\Delta \boldsymbol{y} = \boldsymbol{y}(t) - \boldsymbol{y}(t-1)$ and the image feature error $\tilde{\boldsymbol{y}} = \boldsymbol{y}_d - \boldsymbol{y}$; if $\|\tilde{\boldsymbol{y}}\|=0$, then end; else next step; $\boldsymbol{y}_d$ is desired image feature vector.

4) Using the fuzzy adaptive particle filter to acquire the Jacobian matrix ***J***;

5) Computing the control signal through ***J*** from the image feature error to control the robot moving.

V. EXPERIMENTAL RESULTS

The experiments are executed on a two degree-of-freedom robot system with a CCD camera fixed in the end-effector as in [11]. We use the x-coordinate and y-coordinate in the image plane of the object as image features. The control goal is to keep the robot end-effector tracking the object with the following motion:

$$\begin{cases} x = 0.5 + 0.3\cos(t) \\ y = 0.5 + 0.2\sin(t) \end{cases}$$

The total Jacobian is estimated respectively by traditional particle filter and fuzzy adaptive particle filter, and one set of experiment results is shown in Fig.2, Fig.3 and Fig.4.

In Fig.2, the round point represents an initial position of the robot end-effector, the square point is an initial position of the target, the dotted line is a robot tracking trajectory with traditional particle filter, and the solid line is a robot tracking trajectory with fuzzy particle filters. Fig.3 shows the image feature error curves. Fig.4 shows the robot joint angular changes during the tracking process.

Compared with the experiments of robot visual servo based on traditional particle filter, the proposed method can get comparative tracking accuracy but use less time. Under the same machine environment and with almost the same accuracy, using the proposed robot visual servo method, the running time is cut down from 1.789s to 0.977s.

VI. CONCLUSION

In this paper, a robot visual servo method is proposed, with fuzzy adaptive particle filter to estimate the Jacobian Matrix online. The fuzzy logic idea is introduced into the particle filter framework to adaptively adjust the number of particles. The experiment results show that this proposed visual servo method with fuzzy adaptive particle filter needs less time to get the same accuracy as the method with traditional particle filter. Next we will study more on a real mobile robot system.

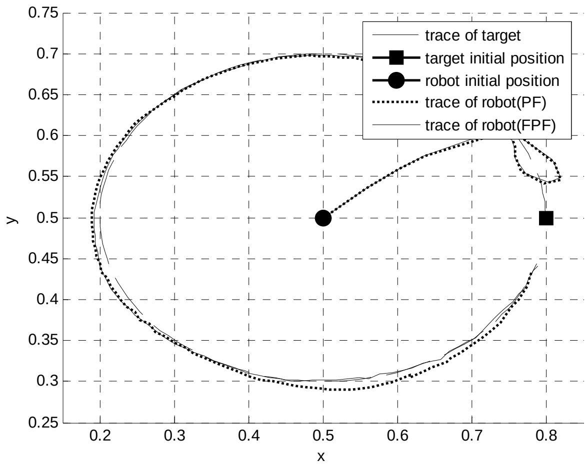

Figure 2. Robot tracking trajectory

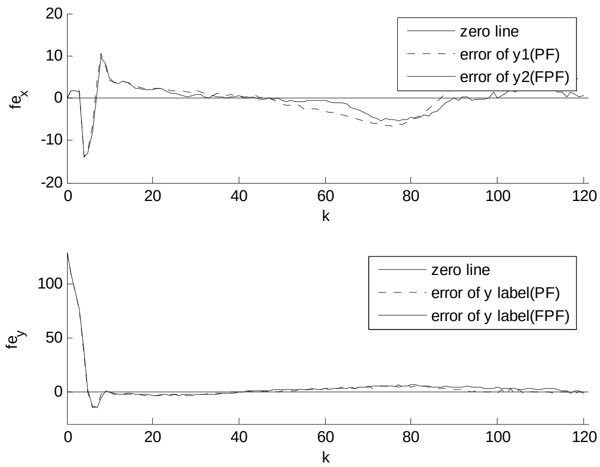

Figure 3. Image feature errors

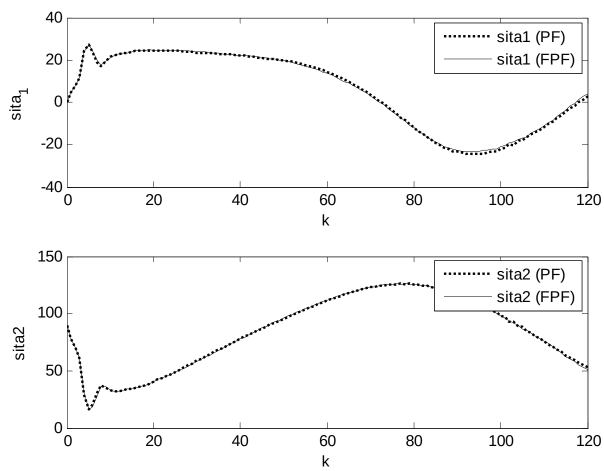

Figure 4. Robot joint angles

REFERENCES

[1] F. Chaumette and S. Hutchinson, "Visual servo control Part I: Basic Approaches," *IEEE Robotics & Automation Magazine*, pp. 82-90, 2006.

[2] M. Hao, *Uncalibrated model free robot visual servoing*, Ph.D thesis, Beijing: Tsinghua University, 2009, in Chinese.

[3] Q. Zhao, *Image-based robot visual servoing*, Ph.D thesis, Beijing: Tsinghua University, 2003, in Chinese.

[4] Q. Zhao, Z. Sun, and H. Deng, "Robot visual servoing based on total Jacobian," Ninth Asian Computing Science Conference, Lecture Notes in Computer Science, LNCS 3321, Springer-Verlag Heidelberg, 2004, pp. 271-285.

[5] K. Hosoda and M. Asada, "Versatile visual servoing without knowledge of true Jacobian," Proceedings of IEEE/RSJ/GI Conference Intelligent Robots and Systems, 1994, pp. 186-193.

[6] M. Jagersand and R. Nelson, "On-line estimation of visual-motor models using active vision," Proceedings of ARPA Image Un-derstanding Workshop, 1996.

[7] J. Piepmeier, G. Mcmurray, and H. Lipkin, "A dynamic quasi-Newton method for uncalibrated visual servoing," Proceedings of IEEE Conference Robotics & Automation, 1999, pp. 1595-1600.

[8] M. Asada, T. Tanaka, and K. Hosoda, "Adaptive binocular visual servoing independently moving target tracking," Proceedings of IEEE Conference Robotics & Automation, 2000, pp. 2076-2081.

[9] J. Qian and J. Su, "On-line estimation of image Jacobian matrix by Kalman-Bucy filter for uncalibrated stereo vision feedback," Proceedings of IEEE International Conference on Robotics and Automation, Vol. 1, 2002, pp. 562 – 567.

[10] X. Lv and X. Huang, "Fuzzy adaptive Kalman filtering based estimation of image Jacobian for uncalibrated visual servoing," IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, 2006, pp. 2167-2172.

[11] Q. Zhao, Y. Chen, and L. Zhang, "Online estimation technique for Jacobian matrix in robot visual servo systems," The 3rd IEEE Conference on Industrial Electronics and Applications, 2008, pp. 1270-1275.

[12] Q. Zhao, Y. Chen, and L. Zhang, "Online estimation of Jacobian matrix based on particle filter," Transactions of Beijing Institute of Technology, 2008, 28(5), pp. 401-404, in Chinese.

[13] N. Gordon D. Salmond, and A. Smith, "Novel approach to nonlinear/non-Gaussian Bayesian state estimation," IEE proceedings-F, vol. 140, No. 2, 1993, pp. 107-113.

JIe Ma received the M.S. degrees the School of Computer Science and Technology, Beijing Institute of Technology, in June, 2010.

Qingjie Zhao received the B.S. and M.S. degrees in Control Science and Technology, from China University of Petroleum in 1988, and Shandong University in 1996, respectively. She received the Ph.D. degree in Computer Science and Technology from Tsinghua University, in 2003.

She is currently a professor in the School of Computer Science and Technology, Beijing Institute of Technology, China. Her current research interests include machine vision, system theory and application. Dr. Zhao is a Member of China Computer Federation.