Target Model Estimation using Particle Filters for Visual Servoing

A. H. Abdul Hafez
Dept. of CSE, College of Engineering
Osmania University
Hyderabad-500007, India
hafezsyr@ieee.org

C. V. Jawahar
Center for Visual Information Technology
International Institute of Information Technology
Gachibowli, Hyderabad-500019, India
jawahar@iiit.ac.in

Abstract

In this paper, we present a novel method for model estimation for visual servoing. This method employs a particle filter algorithm to estimate the depth of the image features online. A Gaussian probabilistic model is employed to model the object points in the current camera frame. A set of 3D samples drawn from the model is projected into the image space in the next frame. The 3D sample that maximizes the likelihood is the most probable real-world 3D point. The variance value of the depth density function converges to very small value within a few iterations. Results show accurate estimate of the depth/model and a high level of stability in the visual servoing process.

1. Introduction

Based on the visual information, visual servoing systems can be classified to three categories [5]: position-based, image-based, and hybrid visual servoing. Position-based visual servoing needs an estimate of the object pose with respect to the camera frame using the complete 3D model of the object. The requirements of the hybrid approaches varies from coarse to accurate estimate of the depth/model. In image based visual servoing, 2D visual information is extracted from the images and used directly in the control law along with the image Jacobian [4] to generate the control signal.

The image Jacobian needs information from the image space and an estimate of the depth of the features points. Consequently, performance of the image-based visual servoing depends on the accuracy of the depth estimate. It was assumed in the literature that a rough estimate of the depth is enough to accurately compute the control signal. Malis and Rives [6] proved analytically and demonstrated that the robustness domain of the image-based visual servoing with respect to depth error is not very wide. They argued that special care should be taken to the depth estimation step for a stable control process. Usually, to avoid an online estimation of the depth, it is done in the desired image and the Jacobian at the desired image is used for the calculation of the control signal. This does not guarantee the image points to be always in the camera field of view (FoV). Indeed, a special care is needed to keep the features in the FoV. Here, we propose a Bayesian method to estimate the depth of the current feature during visual servoing process. Bayesian techniques typically update the state vector using a tracking filter like Kalman filter [3] or particle filter [9].

The Extended Kalman Filter (EKF) has been known as the most common approach in visual tracking applications [2]. The complexity of the EKF grows quadratically with the number of image measurements. The EKF is also very sensitive to outliers in detection of image features. Methods like EKF expect a reasonable initial estimate of the state variables. They often need special care in tuning the noise parameters. In contrast, particle filter deals with outliers better. It does not need any initialization and tuning step. It starts from uniform distribution. However, particle filter performs poorly with respect dimensionality of the state vector. Dimensionality appears as a serious problem in visual simultaneous localization and mapping (VSLAM) applications. However, in visual serving the number of features is limited and fixed. This allows managing the dimensionality probelm satisfactorily.

In this paper, a method that employs the Gaussian particle filter to estimate the depth of the image point online is presented. Initially, we draw particles (samples) of the depth from the visible regions in the current camera frame. These samples are then propagated to the next frame with some level of uncertainty. Sample images provide a likelihood density of the drawn samples. The sample that maximize the density function of the likelihood is assigned to the estimate of the mean of the 3D model distribution. After some iteration the distribution converges to a Gaussian with a sharp peak *i.e.*, a variance value smaller than the threshold set in [6]. One can note that particle filters give an estimates of the full 3D description at the selected feature points.

2. Background and Review

The problem of visual servoing is that of positioning the end-effector of a robot arm such that a set of current features s reaches a desired value s^*. The main objective of the visual servoing process is to minimize the error function $e(s) = s - s^*$. For exponential convergence of the minimization process in a simple proportional control law, the required velocity of the camera can be shown to be [4]

$$V = -\lambda L_S^+ e(s), \tag{1}$$

where $e(s)$ is a $(2N \times 1)$ vector consisting of the errors in image coordinates (u, v) of N points. The velocity $V = (\nu, \omega)^T$ is the camera velocity, ν is translational velocity and ω is rotational velocity. The $(2N \times 6)$ matrix L_s is called the image Jacobian matrix. Image Jacobian relates the changes in the image space to the changes in the Cartesian space [4].

In fact, the stability of visual servoing process is subject to the robustness in image measurements and a good estimate of the depth [6], where these two are included in the Jacobian matrix. The Jacobian matrix can be written as

$$L_s = \frac{1}{Z} A(u, v) + B(u, v), \tag{2}$$

where u and v are the image coordinate vector of all points. One can note that an estimate of the depth is necessary in the camera frame. Most of the recent image-based visual servoing algorithms assume the depth estimate to be available. To the best of our knowledge, there are only limited attempts [7] that takes care of the depth estimation. In that work, they proposed an affine reconstruction method to recover the depth estimation from a pure translation motion as an off-line step in the image-based visual servoing.

3. Bayesian Modeling of Dynamic Systems

Consider a camera mounted on a robot arm manipulator observing an object in the 3D world, the dynamical system is the robot arm and the 3D scene. We assume that the environment is *Markovian*, that is, the past and future data are conditionally independent if the current state is known.

3.1. Object Model Estimation

In object model estimation, the state vector X represent the 3D coordinates of the object points. The measurements data are the image points x correspond to the 3D points X, and the control u, which is commanded to the arm controller. Bayes filters estimate the probability density function over the state space conditionally to the measurements data *i.e.*, image points and control command. This probability is called the *belief* of the state vector and denoted as $\pi(X_t)$. Without loss of generality, we assume that the image measurements and the control commands arrive alternatively. In other words, the control command u_{t-1} is the motion during the time interval $[t-1, t]$ while the current image measurements at the time t is x_t. Based on these assumptions, we can write the belief $\pi(X_t)$ as

$$\pi(X_t) = p(X_t \mid x_t, u_{t-1}, x_{t-1}, u_{t-2}, x_{t-3}, \ldots, x_0). \tag{3}$$

The belief $\pi(X_t)$ is estimated recursively using Bayes filter. The initial belief $\pi(X_0)$ represents the initial knowledge about the system state. In case of there is no *a priori* knowledge about the target model, the initial belief is a uniform distribution over the 3D coordinate space of the object points.

To derive the recursive update equation, we use Bayes rule to write Eq. (3) as

$$\pi(X_t) = \frac{p(x_t \mid X_t, u_{t-1}, \ldots, x_0)\, p(X_t \mid u_{t-1}, \ldots, x_0)}{p(x_t \mid u_{t-1}, \ldots, x_0)}. \tag{4}$$

From *Markov* assumptions and by integrating over the state at time $t-1$, we get the update equation in Bayes filter as

$$\pi(X_t) = \alpha\, p(x_t \mid X_t) \int p(X_t \mid X_{t-1}, u_{t-1})\, \pi(X_{t-1}) dX_{t-1}. \tag{5}$$

Starting from the initial belief or the *a priori* knowledge about the system state, we have a recursive estimator for the object model that is partially observable. To implement the Eq. (5), we need to know the two density functions: (i) the probability $p(X_t \mid X_{t-1}, u_{t-1})$, this is nothing but the motion model of the system, (ii) the density $p(x_t \mid X_t)$, that is the sensor model. One can note that these two models are time invariant and they do not depend on the specific time t. In particle filters the belief $\pi(X)$ is represented by a set of M weighted samples

$$\pi(X_t) \approx \{X_t^m, w_t^m\}_{m=1,\ldots,M}. \tag{6}$$

Here, X_t^m is a *sample* of the random variable X_t, and w_t^m is a non negative parameters called the *importance factors*, these importance factors are normalized in a such a way that they sum upto one. Restating Eq. (5) for the sample pairs, we get

$$\{X_{t-1}^m, X_t^m\} \approx \alpha\, p(x_t \mid X_t^m)\, p(X_t^m \mid X_{t-1}^m, u_{t-1})\, \pi(X_{t-1}^m).$$

Gaussian particle filter operates by approximating the desired densities as Gaussian [1]. Here, only the mean and the variance are propagated along the time to simplify the implimentation. Gaussian particle filter is a Gaussian filter in which particle filter based method is used to estimate the mean and the covariance of the concern densities recursively. Propagation of the mean and the covariance simplify the implementation of the Gaussian particle filter.

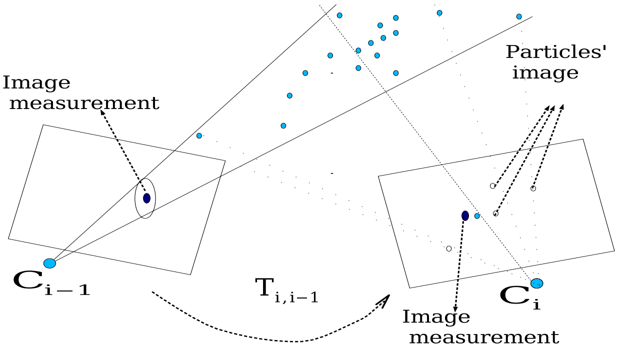

Figure 1. Geometrical description of one step of the particle-based depth estimation.

3.2. Motion Model Estimation

Let the vector $P = (X^c, \dot{X}^c, Y^c, \dot{Y}^c, Z^c, \dot{Z}^c, \theta, \dot{\theta}, \beta, \dot{\beta}, \gamma, \dot{\gamma})$ be the state vector of the motion model, where P is the pose vector of the camera frame related to a reference frame and its derivative with respect to time. The state update and the measurements equations are

$$P_t = AP_{t-1} + w_t, \quad x_t = F(P_t, X_t) + v_t. \tag{7}$$

In these two equations, A is the transition matrix, w_t and v_t are additive noise vectors representing the uncertainty in the motion and the measurements respectively. The covariance matrix of this noise is diagonal since it represent an uncorrelated noise vector. The function F is the measurement function. Using EKF, these prediction equations can be updated recursively as

$$\hat{P}_{t|t} = \hat{P}_{t|t-1} - K_t(x_t - F(P_{t|t-1})), \tag{8}$$

where K_t is the Kalman filter gain. From the pose values $\hat{P}_{t|t}$ and $\hat{P}_{t-1|t-1}$, we compute the transformation $T_{t,t-1}$. Consider the 3D point X_t in the camera frame at the instance t. The point X_{t-1} is mapped to this point in the camera frame through the transformation $T_{t,t-1}$ as $X_t = T_{t,t-1}X_{t-1}$

4. Depth and Object Model Estimation Using Particles

Consider the noisy image point x, corresponding to a world point X. This image point is represented by

$$p(x \mid X) = \frac{1}{2\pi|\Sigma_x|^{1/2}} \exp[(-\frac{1}{2}(x - KX)^T \Sigma_x^{-1}(x - KX))].$$

Here the noise in the image is assumed to be Gaussian with zero mean and variance Σ_x.

If the image points are back-projected with the help of the presently available depth estimate, the distribution of the world points can be characterized by a mean vector $\bar{X}$ and the covariance matrix Σ_X where

$$\bar{X} = [\bar{Z}\bar{u}, \bar{Z}\bar{v}, \bar{Z}]^T, \quad \Sigma_X = J_b^T \begin{pmatrix} \Sigma_x & 0 \\ 0 & \sigma_Z \end{pmatrix} J_b \ . \tag{9}$$

The matrix J_b is the Jacobian of the inverse of the back-projection function. The mean vector is the back-projected coordinate of the mean in the image with the help of the mean-depth. The uncertainty in the 3D configuration is enveloped by a cone that starts from the camera center and passes through the uncertainty ellipse of the image measurements.

When camera moves from the pose P_{t-1} to the pose P_t, the 3D point X_{t-1} will be transformed to X_t using the transformation $T_{t,t-1}$. The uncertainty related to the point distribution gets transformed as

$$\bar{X}_t = T_{t,t-1}\bar{X}_{t-1}, \quad \Sigma_{X(t)} = J_t^T \Sigma_{X(t-1)} J_t. \tag{10}$$

Here, $J_t = \frac{\partial T^{-1}}{\partial X} |_{\bar{X}}$, is the Jacobian of $T_{t,t-1}^{-1}$, which is obtained from the first order approximation.

We draw a set of M 3D points samples (particles) $\{X^m\}_{m=1}^M$ from the density function $p(X_t \mid X_{t-1}) = \mathcal{N}(X_t; \bar{X}_t, \Sigma_{Xt})$ and project it to the image space. The 3D sample point X_t^m, which maximizes the density $p(x_t \mid X_t^m)$ in the current camera frame at the instance t, is assigned to $\bar{X}_t$.

$$\bar{X}_t = \arg\max_{X_t^m}\{w_t^{*(m)} = p(x_t \mid X_t^m)\}, \tag{11}$$

$$\Sigma_{Xt} = \sum_{m=1}^{M} w_t^m (X_t^m - \bar{X}^t)(X_t^m - \bar{X}_t)^T. \tag{12}$$

The normalized weights w_t^m are given by

$$w_t^m = w_t^{*(m)} / \sum_{m=1}^{M} w_t^{*(m)}. \tag{13}$$

By repeating this process recursively from one visual servoing iteration to another, the estimation of the mean $\bar{X}$ becomes accurate and the variance will converge to an acceptable value. Figure (1) depicts the geometrical description of the previous estimation steps. The 3D sample whose image is the nearest to the image measurements is assigned as a mean. In this way, Gaussian particle filter is employed for the estimation of not only the depth distribution but the 3D model. Usually, the depth value which is substituted in the control is obtained by drawing a sample of the estimated 3D distribution.

4.1. Practical Aspects

For the real time model estimation, a boot-strapped technique with few features was used in [2] and [8]. The number of features in visual servoing is limited and there is no meaning for 3D model estimation if we can start with some a priori known features. Instead of boot-strapped, the pose can be initialized using the inverse kinematics of the robot arm and the features using the knowledge of the average distance of the object from the camera. The estimation of the 3D points can be done sequentially. One point at each visual

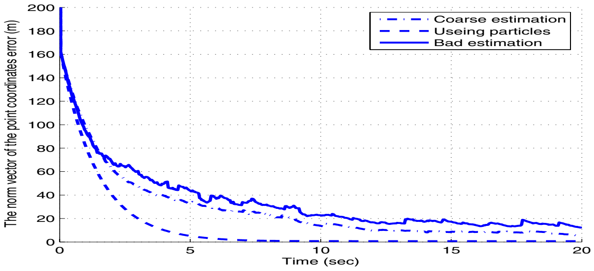

Figure 2. The norm of the feature error vector in case of bad, coarse, and using particle filtering estimation of the depth.

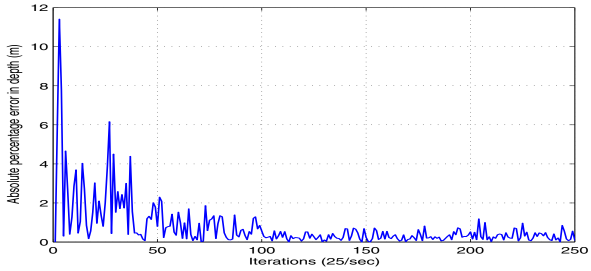

Figure 3. The percentage error in the depth estimation using particles with respect to the real value of the depth.

servoing iteration can be updated. After the first iteration, the estimate of the first point can be used to initialize the remaining points. After the Nth iteration, where N is the number of point, the previous estimate of each 3D point is used in the current frame to estimate the current 3D model.

5. Simulation Results

In the simulation experiments, we use a set of 3D points X_i, $i = 1, ..., N$, for verifying the performance of our algorithm. These points belong to an object in the scene. The task is that the robot arm has to move from initial position to a desired position. The desired position is specified as an image. Coordinates of the image points are considered as features. Since we have N points, there are $2N$ features corresponding to the x and y coordinates in the image. The camera is modeled as a perspective camera with focal lengths $f_x = f_y = 1000m$.

We carried out the experiments for a positioning task using image-based visual servoing. The task is repeated for three different types of depth estimation. In the first case, we assume the depth to be a coarse estimate with noise 10% error in the depth value; In the second case, we used a bad depth estimate, that is 20% error in the depth value. Finally we also consider the depth estimate using the particle filters as discussed in the previous section. Figure (2) compares the error between the image point coordinates versus time in the current and desired position in the image space in the three above cases. It depicts the variation of the norm of the error using particle filters converging to zero while in case of coarse and bad estimation it converge to a two fixed values depends on the amount of error in the depth estimation. Figure (3) shows the absolute percentage error in the depth estimate along the time. One can note that the error converge to around 1% and it is an accurate estimation more than enough as given in [6].

6. Conclusion and Future Work

Estimating the depth distribution using particle filters gives an acceptable results. This increases the stability domain of the visual servoing system with respect to the error in the depth estimation. Estimating the depth gives the estimation of the 3D model. The availability of the depth and the 3D model gives the possibility of of performing either 3D visual servoing, 2D visual servoing, or both. As a future work, integration of 2D and 3D visual servoing to improve the performance of the visual servoing process will be considered.

References

[1] M. Bolic. *Architectures for Efficient Implementation of Particle Filters*. PhD thesis, State University of New York, Stony Brook, USA, 2004.

[2] A. Davison. Real-time simultaneous localization and mapping with a single camera. In *IEEE Int. Conference on Computer Vision, ICCV'03*, 2003.

[3] L. Deng, W. J. Wilson, and F. Janabi-Sharifi. Combined target model estimation and position-based visual servoing. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, IROS'04*, pages 1395–1400, Sendai, Japan, October 2004.

[4] S. Hutchinson, G. Hager, and P. Cork. A tutorial on visual servo control. *IEEE Transactions on Robotics and Automation*, 17:18–27, 1996.

[5] E. Malis, F. Chaumette, and S. Boudet. 2 1/2 d visual servoing. *IEEE Transactions on Robotics and Automation*, 15(2):238–250, April 1999.

[6] E. Malis and P. Rives. Robustness of image-based visual servoing with respect to depth distribution errors. In *IEEE Int. Conf. on Robotics and Automation, ICRA'03*, volume 1, pages 1056–1061, Taipei, Taiwan, Sept. 2003.

[7] E. Malis and P. Rives. Uncalibrated active affine reconstruction closing the loop by visual servoing. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, IROS'03*, volume 1, pages 498–503, Las Vegas, Nevada, U.S.A, October 2003.

[8] M. Pupilli and A. Calway. Real-time camera tracking using a particle filter. In *Proceedings of the British Machine Vision Conference*, pages 519–528. BMVA Press, September 2005.

[9] I. M. Rekleities. A particle filter tutorial for mobile robot localization. Technical Report TR-CIM-04-02, Centre for Intelligent Machines, McGill University, Montreal, Quebec, Canada, 2004.