# Depth Estimation Using Particle filters for Stable Image-based 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 novel method for depth estimation in image-based visual servoing. This method employ the particle filter algorithm to estimate the depth of the image features on-line. A Gaussian probabilistic model is employed to model the object points in the current camera frame. This model is propagated to the next frame. A set of 3D samples drawn from the model is projected into the image space. The maximum likelihood of the 3D samples is the most probable to be the real-world 3D point. The mean and the variance of the model distribution is obtained from the maximum likelihood. The variance values converge to very small value within a few iterations. This gives high level of stability to the visual servoing system.*

## 1 Introduction

Based on the visual information, visual servoing systems can be classified to three methods [8]: position-based, image-based, and hybridvisual servoing. In image based visual servoing, 2D visual information is extracted from image and used directly in the control law along with the image Jacobian [6] to generate the control signal The image Jacobian needs information from the image space and informations about the depth value of the features. Consequently, the image based visual servoing efficiency depends on the depth estimation. In position-based visual servoing, 3D information about the pose of the object frame with respect to the camera frame is estimated. In this case, a complete information about the 3D model of the object is needed. The error function is selected as the difference between the current pose and the desired one of the object frame. Hybrid approaches are different in its dependency on the depth of the features or the object model. The method proposed in [8] and improved in [7] uses information from one point of an image, and works properly for any rough approximation of the depth value of this point. On the contrary, hybrid methods like [3] and the partitioned approach proposed in [2] use the full information available in the image, and as a consequence, they strongly depend on the depth estimation.

It was common in the literature that a rough estimation of the depth is enough to come out with a stable control law in image-based visual servoing. Malis and Rives in [9] proved analytically and using simulation experiments that the robustness domain of image-based visual servoing with respect to depth estimation is not so wide. They agreed that special care should be taken to the depth estimation step for a stable control law.

In this paper a method that employs the Gaussian particle filtering to estimate the depth of the image point on-line. The concept of this method is to draw a particles (samples) of the depth in the current camera frame. These samples are propagated to the next frame with some level of uncertainty and projected to the image. Sample images provide a likelihood density of the drawn samples. The sample that maximize the density function of the likelihood is the most candidate to be the 3D correspondence of the measured image point feature and assigned to the estimate of the mean of the model distribution. The variance of the distribution is computed as a function of the weighted sum of the other samples. After a few iteration the distribution converge to a Gaussian with a sharp peak *i. e.* a variance value smaller than the threshold given in [9]. One can note that particle filters give an estimates of the full 3D object model. Depth information is useful for IBVS and the full 3D model is useful for PBVS. In other words, particle filters provide a general framework to perform 2D and 3D visual servoing or an integration of both.

## 2 Theoretical 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 $S^*$. In image-based visual servoing, the set $S$ can be composed of the coordinates of points that belong to the target object.Consider the error function

The main objective of the visual servoing process is to minimize the error function $e(S)$ that is

$$e(S) = S - S^*. \quad (1)$$

For exponential convergence of the minimization process and using a simple proportional control law, the required velocity of the camera can be shown to be [6]

$$V = -\lambda L_S^+ e(S), \quad (2)$$

where $e(S)$ is a $(2N \times 1)$ error vector between the image coordinates $(u, v)$ of $N$ points. The velocity $V = \frac{dP}{dt} = (v^T, \omega^T)^T$ is the camera velocity, $v$ is translational velocity and $\omega$ is rotational velocity. The pose vector $P = (x, y, z, \alpha, \beta, \gamma)$ is a $(6 \times 1)$ vector. The $(2N \times 6)$ matrix $L_S^+$ is called the pseudo-inverse of the image Jacobian. Image Jacobian relates the changes in the image space to the changes in the Cartesian space [6]. The Jacobian matrix , as shown in [9], can be written as

$$L_s = \frac{1}{Z} A(U, V) + B(U, V), \quad (3)$$

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. Recently in [9], it was shown that the stability range with the depth estimation is not so much wide.

# 3 Baysian Modelings of Dynamic Systems

Bayes filters address the problem of estimating the state vector $X$ of dynamical system using a sensor measurements. 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. Bayes filter assume that the environment is *Markov,* 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 odometry data of the arm are represented by the control $u$ 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 $x_t$. Using this assumption we can write the belief $\pi(X_t)$ as

$$\pi(X_t) = p(X_t \mid x_t, u_{t-1}, x_{t-1}, u_{t-2}, x_{t-3}, \ldots, x_0). \quad (4)$$

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 the case of the estimation of an object 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. (4) as

$$\pi(X_t) = \frac{p(x_t \mid X_t, u_{t-1}, \ldots, x_0)\, p(X_t \mid u_{t-1}, \ldots, x_0)}{p(x_t \mid u_{t-1}, \ldots, x_0)}. \quad (5)$$

By putting the Markov assumptions and integrating over the stae at time $t-1$, we get the update equation in Bayes filter [1]

$$\pi(X_t) = \alpha\, p(x_t \mid X_t) \int p(X_t \mid X_{t-1}, u_{t-1})\, \pi(X_{t-1}) dX_{t-1}. \quad (6)$$

So, starting from Initial belief or a given knowledge about the system state, we have a recursive estimator about the object model that is partially observable. To implement the Eq. (6), we need to know the two density functions: the probability $p(X_t \mid X_{t-1}, u_{t-1})$, this is nothing but the estimation of the next state density or the motion model of the system, and the density $p(x_t \mid X_t)$, that is the sensor model. One can note that the 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

$$\pi(X_t) \approx \{X_t^m, w_t^m\}_{m=1,\ldots,M}. \quad (7)$$

Here, $X_t^m$ is a *sample* of the random variable $X_t$, and $w_t^m$ are a non negative parameters called the *importance factors*, these importance factors are normalized in a such a way that they sum to one. Restating the Eq. (6) for the sample pairs, we get

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

Finally, the non-normalized importance factor $w_t^{*(m)}$ is directly obtained from the probability density of the sensor model as

$$w_t^{*(m)} = p(x_t \mid X_t^m) \quad (8)$$

Gaussian particle filtering operates by approximating the desired densities as Gaussian [1]. In fact, Gaussian particle filtering is a Gaussian filter in which particle filter based method is used to obtain the estimate of 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. Owing to the normalization process, the particle filter is unbiased filter subject to the number of particles. Particle set is Small the bias is more. To determine the suitable number of samples, a technique to set an adaptive number of samples is adopted.

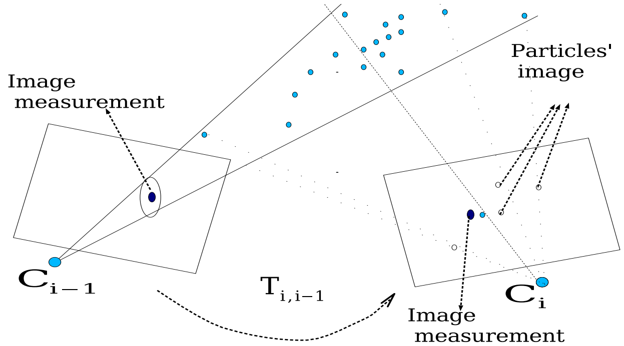


**Figure 1. Geometrical description of one step of the particle-based depth estimation process.**

### 3.2 Motion Model Estimation

In motion model estimation, let us assume that the pose vector of the camera frame related to a reference frame and its derivative with respect to time $P = (X^c, \dot{X}^c, Y^c, \dot{Y}^c, Z^c, \dot{Z}^c, \theta, \dot{\theta}, \beta, \dot{\beta}, \gamma, \dot{\gamma})$ is the state vector of the system. Using the image points and the same control command as measurements, we can write the Bayes recursive update equation as

$$\pi(P_t) = \beta\, p(x_t \mid P_t) \int p(P_t \mid P_{t-1}, u_{t-1})\, \pi(P_{t-1}) dP_{t-1}.$$

The state space equation and the measurements equation are [4]

$$\begin{aligned} P_t &= AP_{t-1} + w_t, \\ x_t &= F_t(P_t, X_t) + v_t. \end{aligned} \tag{9}$$

In these two equations $A$ is the transition matrix,$w_t$ and $v_t$ are additive noise vectors represent the uncertainty in the motion and the measurements respectively. The function $F_t(P, X_t) = KP_w^c X^w$ is the measurement function, where $K$ is the projective camera matrix. Using extended Kalman filter, these prediction equations can be updated recursively by

$$\hat{P}_{t|t} = \hat{P}_{t|t-1} - K_t(x_t - F(P_{t|t-1})), \tag{10}$$

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 $(t)$. This 3D point is mapped to the point $X_{t-1}$ in the camera frame at the previous time instance $(t-1)$ through the transformation $T_{t,t-1}$ as $X_t = T_{t,t-1} X_{t-1}$

## 4 Depth Estimation Using Particles

This section describes how Gaussian particle filtering can be used for depth or model estimation in visual servoing. The estimation process draws samples from the depth density in the current iteration for a selected point features, then predict the correspondences to these samples in the next iteration. Projecting these samples into the next image produces a likelihoods which are used to estimate mean and variance of the updated density of the depth. After a few iterations, the variance will converge to a suitable value and the mean to the real value of the depth.

Consider the image point $x = (u, v)$, which is a projective of the 3D point $X$. The measurement $x$ of this image point can be corrupted by noise and errors. This corruption can be represented by a Gaussian distribution with zero mean and variance $\Sigma_x$. This will result in a random variable with probability distribution [5] given by

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

Given a probability distribution $p(Z) = \mathcal{N}(Z; \bar{Z}, \sigma_Z)$ of the depth, the uncertainty in the image measurements can be back-projected to the Cartesian space using the function $F^{-1}$. A probability density function of the 3D point $X$ that corresponds to the image point measurement $x$ is obtained. This function is $p(X \mid x) = \mathcal{N}(X; \bar{X}, \Sigma_X)$ and the parameters $\bar{X}$ and $\Sigma_X$ are computed as follows [5]

$$\left\{ \begin{array}{l} \bar{X} = [\bar{Z}\bar{u}, \bar{Z}\bar{v}, \bar{Z}]^T \\ \Sigma_X = J^T \begin{pmatrix} \Sigma_x & 0 \\ 0 & \sigma_Z \end{pmatrix} J \end{array} \right. \tag{11}$$

The matrix $J$ is the Jacobian of the inverse of the back-projection function and defined as $J = \frac{\partial F}{\partial X}\Big|_{\bar{X}}$. 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 $T_{t,t-1}$. In case of a probabilistic model of the 3D point, the transformed uncertainty is given as

$$\left\{ \begin{array}{l} \bar{X}_t = T_{t,t-1}(\bar{X}_{t-1}) \\ \Sigma_{X(t)} = J^T \Sigma_{X(t-1)} J \end{array} \right. \tag{12}$$

where $J = \frac{\partial T^{-1}}{\partial X} \mid_{\bar{X}}$, is the Jacobian of the $T_{t,t-1}^{-1}$ which is obtained from the first order approximation.

Let us 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 getting the particles $\{x_t^m\}_{m=1}^M$. To estimate the parameter vector $X_t$ given the measurement $x_t$, we define the function $\bar{X} = \hat{X}_t(x_t)$. This function assigns a 3D point sample $X_t^m$ to the measurement $x_t$, where this 3D point maximize the density $p(x_t \mid X_t^m)$ in the current camera frame at the instance $t$.

$$\bar{X}_t = \hat{X} = \arg\max_{X_t^m} \{w_t^{*(m)} = p(x_t \mid X_t^m)\}, \tag{13}$$

$$\Sigma_{Xt} = \sum_{m=1}^{M} w_t^m (X_t^m - \bar{X}^t)(X_t^m - \bar{X}_t)^T. \tag{14}$$

The normalized weights $w_t^m$ are given by

$$w_t^m = w_t^{*(m)} / \sum_{m=1}^{M} w_t^{*(m)}. \tag{15}$$

By repeating this process recursively from one visual servoing iteration to another, the estimation of the mean $\bar{X}$

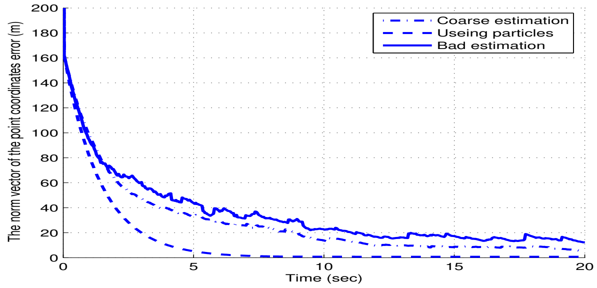


**Figure 2. The norm of the fearture error vector in case of bad, coarse, and useing particle filtering estimation of the depth.**

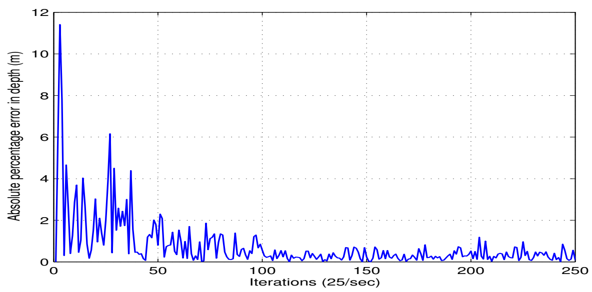


**Figure 3. The percentage error in the depth estimation useing particles with respect to the real value of the depth.**

will be more accurate and the variance will converge to an acceptable value. Figure (1) shows the geometrical description of the previous estimation steps. In this way Gaussian particle filter is employed in this fashion to 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.

## 5 Simulation Results

In the simulation experiments, we used 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 given as a desired image of the object. The image point coordinates are considered as features. Since we have $N$ points, the total number of features is $2N$. In other words, features $S_{2i}$ and $S_{2i-1}$ correspond to the point $x_i$. The camera is modeled as a perspective camera with focal lengths $f_x = f_y = 1000$ m and zero aspect ratios. The dimensions of the image are $512 \times 512$.

We carried out the experiments for a positioning task using image-based visual servoing. Tha task is repeated for a coarse, bad, and using particle filters depth estimation. Figure (2) shows a comparison of the error between the image point coordinates in the current and desired position in the image space. It shows how 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 depnd on the amount of error in the depth estimation. Figure (3) shows the absolute percentage error in the depth estimation along the time. One can note that it converge to around 1% and that is an accurate estimation more than enough as given in [9].

## 6 Conclusion and Future Work

It was proved that the stability domain with respect to the depth value is not so wide. Estimating the depth distribution using particle filters gives a fine estimation of the depth, that increases the stability domain of the 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] P. Corck and S. Hutchinson. A new partitioned approach to image -based visual servoing. *IEEE Transactions on Robotics and Automation*, 14(4):507–515, Aug 2001.

[3] K. Deguchi. Optimal motion control for image-based visual servoing by decoupling translation and rotation. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, IROS'98*, volume 2, pages 705–711, October 1998.

[4] 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.

[5] G. Flandin and F. Chaumette. Visual data fusion: Application to object localization and exploration. Technical Report INRIA/RR-4168-FR+ENG, INRIA, Renne, France, 2001.

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

[7] E. Malis and F. Chaumette. Theoretical improvements in the stability analysis of a new class of model-free visual servoing methods. *IEEE Trans. on Robotics and Automation*, 18(2):176–186, April 2002.

[8] 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.

[9] 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.