Depth Estimation Using Particle filters for Image-based Visual Servoing

A.H. Abdul Hafez *

* *Dept. of Computer Engineering, Faculty of Engineering,*
Hasan Kalyoncu University, Sahinbey, 27410 - Gaziantep, Turkey
(e-mail: abdul.hafez@hku.edu.tr).

Abstract: In this paper, we present a novel approach for depth estimation in image-based visual servoing. Depth information are directly used in the control law to generate control signal, *i.e.* the screw velocity of the robot end-effector. Because rough estimates of depth values are not enough, we are motivated to this proposal. This approach employs a particle filter algorithm to estimate the depth of the image features online. A Gaussian probabilistic model is employed to model depth distribution. A set of depth particles is drawn in the current camera frame. The image measurements are used to recover the 3D samples. These samples are propagated to the next frame and projected into the image space. The maximum likelihood of 3D samples is the most probable to be the real-world 3D point. The mean value and the variance of the depth distribution are 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 image-based visual servoing system. The simulation experiments show that the mean value goes very close to the real value of the depth in a few iterations. The depth is considered as the mean value of estimated distribution.

Keywords: Visual servoing; Particle filter; Depth estimation.

1. INTRODUCTION

Since vision provides non-contact measurements of the environment, cameras are considered as a useful sensor for robot applications. Shirai and Inoue demonstrated in their seminal work on robot vision control (Shirai and Inoue (1973)), that closing the control loop of the robot using visual feedback has more effect than increasing the accuracy of the individual parts of the robot vision system (Borangiu (2004); Corke (2011)). Using visual feedback to close the loop of position control of the end effector of robot arm is now referred to as *visual servoing*. This is achieved by processing a visual feedback and minimizing an appropriate error function. The visual feedback can be image features (2D) or object pose (3D) with respect to the camera frame.

Visual servoing systems can be classified, based on the visual information used in the control law, into three methods, i.e. position-based, image-based, and hybrid (or $2\frac{1}{2}$D) visual servoing. In image based visual servoing, 2D visual information are extracted from both image space and depth map. In position-based visual servoing, 3D information about the pose of the object frame with respect to the camera frame are estimated. In this case, complete information about the 3D model of the object are needed. The error function is selected as the difference between the current pose and the desired one of the object frame. In contrast, Image-based visual servoing computes the velocity from the error function of the image space features and the image Jacobian (Hutchinson et al. (1996); Chaumette and Hutchinson (2006); Abdul Hafez (2014)). Image Jacobian needs information from both image space and depth map. Therefore, it is crucial to use depth estimate to boost the image based visual servoing efficiency.

Hybrid approaches are different in the manner in which image space information are used in the control law. The method proposed in (Malis and Chaumette (2002)) uses information from one point of an image, and works properly for any rough approximation of the depth value of this point. On contrary, hybrid methods like (Deguchi (1998)) and the partitioned approach proposed in (Corke and Hutchinson (2001)) use the full information available in the image, and as a consequence, they strongly depend on the depth estimates. Stability analysis of image-based and hybrid methods except the one proposed in (Malis and Chaumette (2002)) depends on the depth estimation accuracy.

It was common in the literature that a rough estimate of the depth is enough to come out with a stable control law in image-based visual servoing. Malis and Rives proved analytically and using simulation experiments in (Malis and Rives (2003a)) 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. Later, they proposed in (Malis and Rives (2003b)) an affine reconstruction method to recover the depth from a pure translation motion as an off-line step in the image-based visual servoing.

In this paper, a method that employs a 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 maximizes the density function of the likelihood is the most probable candidate to be the 3D correspondence of the measured image feature. This point is then assigned to the estimate of the mean of the model distribution. The variance is the weighted sum of the distances of the mean to other samples. After some iteration the distribution converges to a Gaussian with a sharp peak *i. e.*, a variance value smaller than the threshold set in Malis and Rives (2003a). One can note that particle filters give an estimates of the full 3D description at the selected feature points.

There are several works in the literature that consider the problem of depth (or structure) estimation. For example, the work proposed by Collewet and Chaumette in (Collewet and Chaumette (2008)) focuses on the way to achieve accurate visual servoing tasks when the shape of the object being observed as well as the desired image are unknown. Another example is the IBVS scheme which is proposed in (Chen et al. (2006)) for a camera mounted on a nonholonomic mobile robot via an on-line estimation of a constant unknown parameters, *i.e.* the height of the object points and the depth of the target plane at the desired pose, respectively. The reconstruction phase is based on the measurement of the 2D motion in a region of interest and on the measurement of the camera velocity. The work presented in (Luca et al. (2007)) recasts the problem into the nonlinear observer framework, which provides techniques to estimate unmeasurable time-varying states of known dynamical systems. More recently, a method was presented in (Mao et al. (2012)) to estimate the image Jacobian for visual serving process. Estimates of the Jacobian's elements include the depth of the features. Authors utilise on-lines support vector regression to estimate the depth and other parameters of the servoing process. An active estimation strategy is proposed in (Robuffo Giordano et al. (2014)) in which a monocular camera tries to determine whether a set of observed point features belongs to a common plane, and what are the associated plane parameters. It is proposed in (Abdul Hafez and Cervera (2014)) to use particle filter to estimate the relative motion of the camera. This motion is represented by the relative pose between the initial and current cameras. They showed that motion estimation is accurate and can be used for structure computation.

Particle filters play an important role in variable tracking for robotics. There are many real-time implementation of these algorithms (Davison (2003)). A tutorial on particle filters for mobile robot localization (Rekleities (2004)) discusses a variety of computational and conceptual issues related to these algorithms. An adaptive real-time particle filters for robot localization is presented in (Kwok et al. (2003)).

Our method employs Gaussian particle filtering for online estimation of the depth of the image point. The concept of this method is to draw particles (samples) from the depth distribution 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 maximizes the density function of the likelihood is the most probable candidate to be the depth of the measured image point feature. This value is assigned to the estimate of the mean of the depth distribution. The variance of the distribution is computed as the weighted sum of the squared difference of the samples from the mean. After a few iteration the distribution converges to a Gaussian density function with a sharp peak *i. e.* a variance value smaller than an accepted threshold. The current estimate of the distribution is used for drawing a set of particles again. The process is repeated in the next iteration.

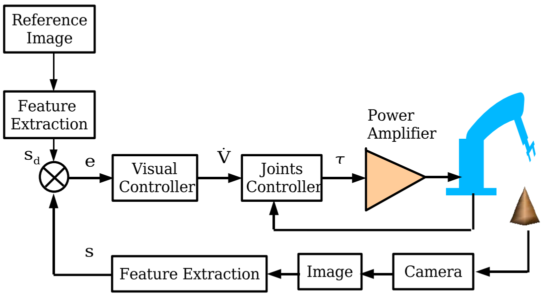


Fig. 1. Block diagram for Dynamic look-and-move visual servoing system.

2. THEORETICAL BACKGROUND

2.1 Projective Camera Model

A camera maps the 3D world to a 2D image. A general projective camera is represented by an arbitrary homogeneous (3×4) matrix of rank 3. The general projective camera K maps world points X to image points $\mathbf{x}$ according to $x = f(X) = KX$. The matrix K includes *internal parameters*, *i.e.* the camera's focal length and the skew angle, and the *external parameters* which specify the camera's position and orientation in the world Hartley and Zisserman (2003). As the matrix K is invertible, the function $f^{-1}(x)$ exists. The 3D coordinates of the point X can be recovered, and hence the depth Z is extracted.

2.2 Control Schemes in Visual Servoing

The term "visual servoing" was introduced by Sanderson and Weiss in (Sanderson and Weiss (1980)). Their taxonomy poses two fundamental concerns:

(1) Is the vision system providing input to the robot's joint-level controller, or does the visual controller directly compute the joint-level inputs?
(2) Is the error signal defined in 3D Cartesian space or directly in 2D image space?

Addressing each of the above concerns, provides a method of classifying the vision-based robot control systems. The following classification arises from the first concern:

(1) Dynamic look-and-move systems: As illustrated in Figure 1, these systems provide a set-point as input to the joint-level robot controller. Then, internal controller uses the joint feedback to stabilize the robot arm and regulate the joint to the desired value that is the set-point provided by the visual controller.
(2) Direct visual servoing systems: As illustrated in Figure 2, these systems use vision alone to stabilize the arm. The visual servo controller directly compute the joint inputs. The visual servo controller here eliminates the internal robot controller.

As pointed out by Hutchinson *et al.* (Hutchinson et al. (1996)), nearly all of the proposed systems adopt the dynamic look-and-move approach. This is mainly due to two reasons: (i) the low sampling rate available from vision system makes direct control of the robot's joints more complex, and (ii) most of robot systems already have an interface for accepting Cartesian velocity or incremental position command. However, the 1 ms hierarchical vision system, presented in 2003 as a high speed

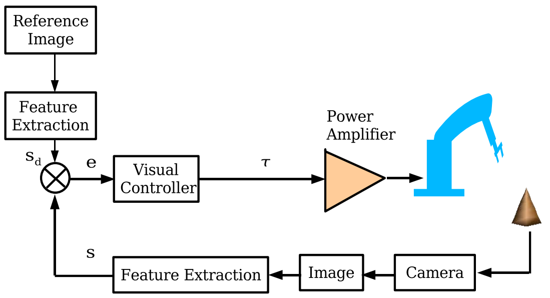


Fig. 2. Block diagram for Direct visual servoing system.

visual servoing system by Namiki *et al.* (Namiki et al. (2003)) uses the direct visual servoing system to control the joint of the arm.

To avoid the confusion of the term visual servoing introduced by Sanderson and Weiss (Sanderson and Weiss (1980)), one may pay attention to the fact that the term visual servoing now is widely used to describe all types of closed-loop vision robot control systems, including the dynamic look-and-move systems (Hutchinson et al. (1996)).

Addressing the second concern shows the three main classes based on whether the error is represented in the 3D space or in the image space.

(1) Position-based Visual Servoing (PBVS): In this approach, the features are extracted from the image and used in conjunction with a CAD model of the target object to know the position and direction (pose) of the object with respect to the camera.
(2) Image-based Visual Servoing (IBVS): In this approach, The features are extracted from the image and used directly to estimate the control signal using *image Jacobian*. This approach may reduce the errors due to sensor modeling and camera calibration, but it presents a significant challenge since the resulting system is nonlinear and highly coupled.
(3) Hybrid Visual Servoing: Hybrid systems combine the two previous approaches. The error to be minimized is specified both in the image and pose space.

Reader may refers to Abdul Hafez (2014) for more detailed information about the three main classes Image-based Visual Servoing (IBVS), Position-based Visual Servoing (PBVS), and Hybrid Visual Servoing.

2.3 Image-based Visual Servoing

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. Other kind of geometric features like straight line segments, angles, or spheres can be also used. Consider the error function

$$e(S) = S - S^*. \tag{1}$$

where S is a vector represents the current set of features and S^* is a vector represents the desired set of features.

By differentiating this error function with respect to time, with the desired features S^* remaining constant, we get

$$\frac{de}{dt} = \frac{dS}{dt} = (\frac{\partial S}{\partial P})\frac{dP}{dt} = L_S V, \tag{2}$$

where $e(S)$ is a $(2N \times 1)$ error vector between the image coordinates (u, v) of N points. The velocity

$$V = \frac{dP}{dt} = (v^T, \omega^T)^T \tag{3}$$

is the camera velocity, v is translational velocity and ω is rotational velocity. The pose vector

$$P = (x, y, z, \alpha, \beta, \gamma) \tag{4}$$

is a (6×1) vector. For exponential convergence of the minimization process, i.e.

$$\frac{de(S)}{dt} = -\lambda e(S), \tag{5}$$

and using a simple proportional control law, the required velocity of the camera can be shown to be (Hutchinson et al. (1996))

$$V = -\lambda L_S^+ e(S). \tag{6}$$

The $(2N \times 6)$ matrix L_S is called the image Jacobian. Image Jacobian relates the changes in the image space to the changes in the Cartesian space (Hutchinson et al. (1996)).

Assuming a perspective projection model with a unit focal length, the interaction matrix L_{S_i} for each point (u, v) is given by Hutchinson et al. (1996):

$$L_{S_i} = \begin{bmatrix} \frac{-1}{Z} & 0 & \frac{u}{Z} & uv & -(1+u^2) & v \\ 0 & \frac{-1}{Z} & \frac{v}{Z} & 1+v^2 & -uv & -u \end{bmatrix}. \tag{7}$$

For a set of N points, the set of features is S_i, $i = 1, ..., N$, the interaction matrix L_S is

$$L_S = \left[\, L_{S_1}, \dots, L_{S_N} \,\right]^T, \tag{8}$$

where L_{S_1} and L_{S_N} are the interaction matrices given in (7) and correspond to points 1 and N respectively. The Jacobian matrix, as shown in (Malis and Rives (2003a)), can be written as

$$L_s = \frac{1}{Z} A(U, V) + B(U, V), \tag{9}$$

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 for image-based visual servoing.

It was assumed that a rough estimation of the depth is enough for a stable control law in image-based visual servoing. However, it was shown in (Malis and Rives (2003a)) that the stability range with respect to the depth estimation is not so much wide.

2.4 Dynamic State Model of 3D Point in Visual Servoing System

The velocity V computed in equation (6) is the control input to robot arm controller. The actual velocity $\hat{V}$ that represents the uncertainty and delay in robot arm dynamics can be represented by adding a term of noise to the computed velocity V.

$$\hat{V} = V + \mathcal{N}(\mu_v, q\mathbf{I}_{6\times6}) = (\hat{v}^T, \hat{\omega}^T)^T,$$

where μ_v is a (6×1) mean vector, and q is a random variable. Let the velocity at the time instance t be $\hat{V}_t$. This velocity will drive the robot arm from the pose P_t at the time instance (t) to the pose P_{t+1} at the time instance $(t + 1)$. The transformation $T_{t+1,t} = (R, \mathbf{t})$ between the current pose and the pose at the next time instance is computed from the velocity as

$$R = I_{3\times3} + [\omega]_\times.\Delta t, \tag{10}$$

$$\mathbf{t} = v.\Delta t, \tag{11}$$

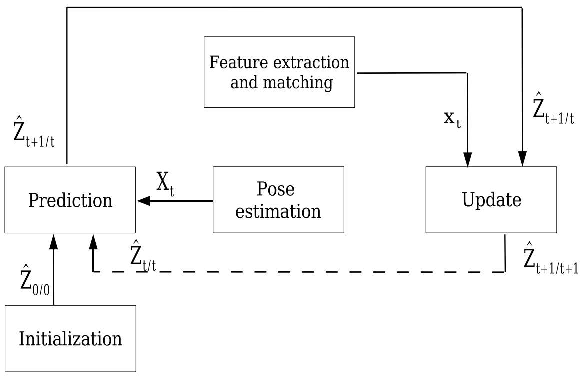


Fig. 3. Block diagram of the model estimation. The prediction and update stages are the core of the pose/model algorithm

where Δt is the time interval between successive visual servoing iterations.

Let us consider the 3D point $\mathbf{X_t}$ in the instance frame of a perspective camera. This point is projected to the image point $\mathbf{x_t}$ using the the camera matrix K. This 3D point $\mathbf{X_t}$ is mapped to the point $\mathbf{X_{t-1}}$ in the camera frame at the previous time instance $(t-1)$ through the transformation $T_{t,t-1}$ as

$$\mathbf{X_t} = T_{t-1,t}\mathbf{X_{t-1}}. \tag{12}$$

This is the dynamic model of the 3D point motion in visual servoing.

3. BAYESIAN OBJECT MODEL ESTIMATION FROM KNOWN POSE

3.1 Model State Vector

In object model estimation, as it is shown in Figure 3, the state vector X represents the 3D coordinates of the object points. The measurement data are the image points x corresponding to the 3D points X and the odometry data of the arm represented by the control u commanded to the arm controller. Bayes filters estimate the probability density function over the state space conditionally to the measurement data *i.e.* image points and control command. This probability is called the *belief* of the state vector and denoted as $\pi(X_t)$.

$$\pi(X_t) = p(X_t \mid x_{0\ldots t}, u_{0\ldots t}). \tag{13}$$

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, the belief $\pi(X_t)$ can be written as

$$\pi(X_t) = p(X_t \mid x_t, u_{t-1}, x_{t-1}, u_{t-2}, x_{t-3}, \ldots, x_0). \tag{14}$$

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. This initial knowledge is given by a probability function computed using a given 3D model of the scene. This function is assumed to be uniform in the absence of any initial knowledge. In 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 Equation (14) as

$$\pi(X_t) = \frac{p(x_t \mid X_t, u_{t-1}, \ldots, x_0)\, p(X_t \mid u_{t-1}, \ldots, x_0)}{p(x_t \mid u_{t-1}, \ldots, x_0)}. \tag{15}$$

By employing the *Markov* assumptions and integrating over the state X_{t-1} at time $t-1$, we get the update equation in Bayes filter (Bolic (2004)) as

$$\pi(X_t) = \alpha\, p(x_t \mid X_t) \int p(X_t \mid X_{t-1}, u_{t-1})\, \pi(X_{t-1}) dX_{t-1}. \tag{16}$$

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 Equation (16), we need to know the two density functions: the probability $p(X_t \mid X_{t-1}, u_{t-1})$, which is nothing but the estimate of the next state density or the motion model of the system as in Eq (12), and the density $p(x_t \mid X_t)$, which is the sensor model.

In particle filters the belief $\pi(X)$ is represented by a set of M weighted samples $\{X_t^m\}_{m=1}^M$,

$$\pi(X_t) \approx \{X_t^m, w_t^m\}_{m=1,\ldots,M}. \tag{17}$$

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

$$w_t^{*(m)} = p(x_t \mid X_t^m). \tag{18}$$

The normalized importance factors $w_t^{*(i)}$ are computed in such a way that its summation is equal to 1.

Let us consider the case of estimating a Gaussian distribution function. This is usually referred to as Gaussian particle filtering. It operates by approximating the desired densities as a Gaussian (Bolic (2004)). Here, only the mean and the variance are propagated along the time. 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 concerned densities recursively. Propagation of the mean and the covariance simplifies the implementation of Gaussian particle filter. Owing to the normalization process, particle filter is an unbiased filter subject to the number of particles. The smaller particle set, the higher the bias is. To determine the suitable number of samples, a technique which sets an adaptive number of samples is adopted. This technique observes the summation of the non normalized weights. When this summation exceeds a certain threshold, the number of samples is decreased.

3.2 Model Estimation and Uncertainty Propagation Using Particles

This subsection describes how Gaussian particle filtering can be used for depth or model estimation from controlled motion. 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 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 will converge to the real value of the depth.

Consider the image point $x = (u, v)$, which is a projection 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 Σ_x. This results in a random variable with probability distribution (Flandin and Chaumette (2001)) given by

$$p(x \mid X) = \frac{1}{(2\pi|\Sigma_x|^{1/2})}\exp[(-\frac{1}{2}(x-KX)^T\Sigma_x^{-1}(x-KX))]. \tag{19}$$

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 Σ_X are computed as follows Flandin and Chaumette (2001)

$$\bar{X} = [\bar{Z}\bar{u}, \bar{Z}\bar{v}, \bar{Z}]^T,\ \Sigma_X = J_F^T\begin{pmatrix}\Sigma_x & 0\\ 0 & \sigma_Z\end{pmatrix}J_F\,. \tag{20}$$

Here, the matrix J_F is the Jacobian of the inverse of the back-projection function (Flandin and Chaumette (2001)) and defined as

$$J_F = \frac{\partial F^{-1}}{\partial X}\Big|_{\bar{X}} = \begin{pmatrix}1/\bar{Z} & 0 & -\bar{u}/\bar{Z}\\ 0 & 1/\bar{Z} & -\bar{v}/\bar{Z}\\ 0 & 0 & 1\end{pmatrix}. \tag{21}$$

The parameters of the 3D point distribution are given by a set of M samples (particles). These samples can be drawn of the 3D points $\{X\}_{m=1}^{M}$, which is recovered completely from the image measurement x and the function $F^{-1}(x)$. 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

$$\bar{X}_t = T_{t,t-1}(\bar{X}_{t-1}),\ \Sigma_{X(t)} = J^T\Sigma_{X(t-1)}J, \tag{22}$$

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

Let us draw a set of M 3D point samples (particles) $\{X^m\}_{m=1}^{M}$ from the density function $p(X_t \mid X_{t-1}) = \mathcal{N}(X_t; \bar{X}_t, \Sigma_{Xt})$, as it shown in Figure 4, 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 maximizes the density $p(x_t \mid X_t^m)$ in the current camera frame at the instance t. In fact, this function is nothing but the maximum likelihood of the density $p(x_t \mid X_t^m)$ and is written as

$$\bar{X}_t = \hat{X} = \arg\max_{X_t^m}\{p(x_t \mid X_t^m)\}, \tag{23}$$

$$\Sigma_{Xt} = \sum_{m=1}^{M} w_t^m(X_t^m - \bar{X}^t)(X_t^m - \bar{X}_t)^T. \tag{24}$$

The 3D point $\hat{X}$ is assigned to the mean of the new distribution $\bar{X}$, while the variance Σ_{Xt} will be computed using a set of weights proportional to the density values $p(x \mid X^m)$ along the samples $\{X^m\}_{m=1}^{M}$. The normalized weights w_t^m are given by

$$w_t^m = w_t^{*(m)} / \sum_{m=1}^{M} w_t^{*(m)}. \tag{25}$$

These weights $w_t^{*(m)}$ are the likelihoods of the samples X_t^m with respect to the measurement x_t and computed as

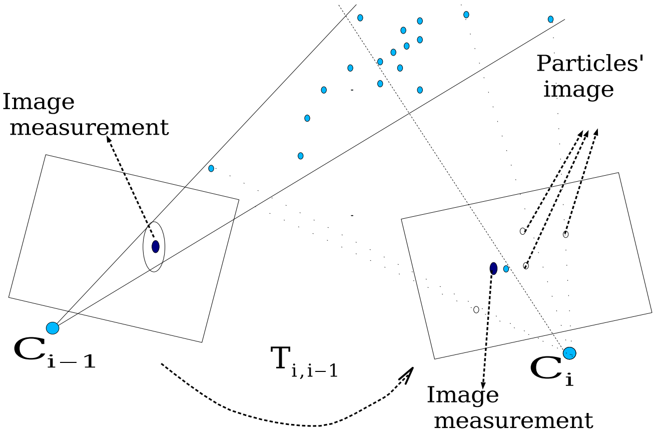


Fig. 4. Geometric description of one step of the particle-based depth estimation process.

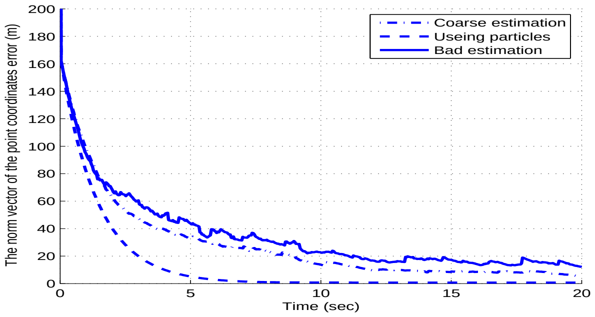


Fig. 5. The norm of the error in the image coordinates vector during a visual servoing process. This error should be regulated to zero. Three methods for depth estimation are shown. Two of them, that produce coarse and bad estimates, are compared to the methods using particle filtering.

$$\begin{cases} w_t^{*(m)} = p(x_t \mid X_t^m)\\ w_t^{*(m)} = (2\pi\sigma^2)^{-1/2}\exp(-(d_t^m)^2/2\sigma^2), \end{cases} \tag{26}$$

where d_t^m is the distance between the image measurement point x_t and the projection of X_t^m that is the particle m at the time instance t. The variance σ is a function of the image noise variance. By repeating this process recursively from one visual servoing iteration to another, the mean $\bar{X}$ and variance converge to an accurate value. Figure (4) shows 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. The steps of the algorithm are summarized in Algorithm 1

4. RESULTS AND DISCUSSION

We conduct our experiments in simulation platform built using Matlab environment (Corke (2011)). The expected disturbances like image measurement error, tracking error, calibration error are properly modeled in the considered simulation platform. The considered system is eye-in-hand visual servoing system where the camera is mounted on the robot arm and observing only the target object. This system is called Endpoint open-loop (EOL) (Hutchinson et al. (1996)). Even though we imple-

Algorithm 1 Model/depth target estimation algorithm from visual servoing motion using particle filter.

```
Input:
    $p(x_{t-1})$                            % The measurement density at time $t-1$ in the previoous image.
    $p(x_t)$                                % The measurement density at time $t$ in the current image.
    $\bar{X}_{t-1} = \mu_{X_{t-1}}$         % The mean of the depth distribution at time $t-1$.
    $\Sigma_{X_{t-1}}$                      % The variance of the depth distribution at time $t-1$.

Output:
    $\bar{X}_t$                             % The mean of the updated belief of the 3D point depth.
    $\Sigma_{Xt}$                           % The variance of the updated belief of the 3D point depth.

Prediction stage
    $p(X_{t-1} \mid x_{t-1}) = \mathcal{N}(X_{t-1}; \bar{X}_{t-1}, \Sigma_{X(t-1)})$    % Using the back-projection in Equation 20.
    Compute $p(X_t \mid X_t = X^m_{t-1}, x_t)$                                         % Using the transformation $T_{t,t-1}$
                                                                                       that is the estimate of the relative camera pose.
    Draw the particles $\{X^m_t\}^M_{m=1}$                                             % Using the previous depth distribution
                                                                                       $\mathcal{N}(X^m_{t-1}; \mu_{X_{t-1}}, \Sigma_{X_{t-1}})$

Update stage stage
    Generate the image particles $\{x^m_t\}^M_{m=1}$                                   % By projecting the particle
                                                                                       $\{X^m_t\}^M_{m=1}$ to the image space at time $t$.
    Calculate the weights $w^{*(m)}_t = p(x_t \mid X^m_t)$                             % As in Equation (26).
    Normalize the weights $w^m_t = w^{*(m)}_t / \sum^M_{m=1} w^{*(m)}_t$               % To form a distribution.
    Compute the updated estimates:
    $\bar{X}_t = \hat{X} = \arg\max_{X^m_t}\{w^{*(m)}_t = p(x_t \mid X^m_t)\}$,        % The mean of the model distribution
    $\Sigma_{Xt} = \sum^M_{m=1} w^m_t (X^m_t - \bar{X}^t)(X^m_t - \bar{X}_t)^T$        %The variance of the model distribution

return $\bar{X}_t$ , $\Sigma_{Xt}$
```

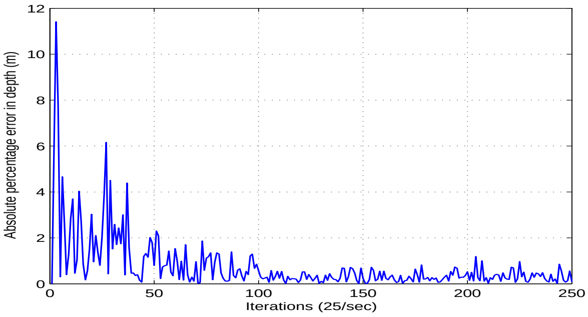


Fig. 6. The percentage error in the depth estimates, during visual servoing process, using particles with respect to the real value of the depth.

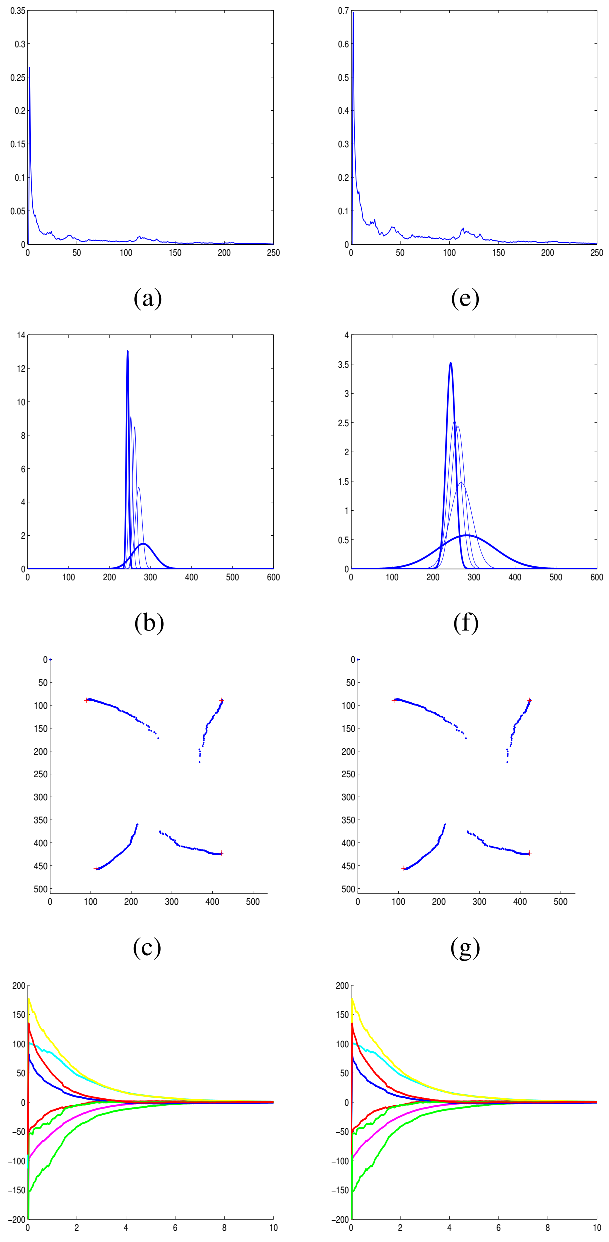
mented our particle filter proposal using Matlab, we believe that it runs at frame rate not less that 15 frame/second speed using C/C++ particle filter implementation. See for example results reported from our C++ based tracking and servoing work using particle filter (Abdul Hafez and Cervera (2014)).

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 $2N$. With assumption that enough number of features are extracted and tracked, there is no considerable importance for the remaining parts/shape of the object. The camera is modeled as a perspective camera with focal lengths $f_x = f_y = 1000m$, unit aspect ratio, and zero skew. Dimensions of the images is 512×512 pixels. Since we are working using simulation framework, we do not have camera calibration problem here which is usually solved as in (Abdul Hafez and Cervera (2014)).

We conducted experiments for a positioning task using image-based visual servoing. The task is repeated for three different depth estimations, coarse, bad, and using particle filters depth estimation. Figure 5 shows a comparison of the error between the image point coordinates in the image space. It shows how the norm of the error using particle filters converges to zero while in case of coarse and bad estimation it converges to two fixed values depending on the amount of error in the depth estimation. Figure 6 shows the absolute percentage of error in the depth estimation along the time. One can note that it converges to around 1% and that is an accurate estimate as mentioned in Malis and Rives (2003a).

4.1 Sensitivity to image noise

The first experiment is carried out with four points from a non-planar object. Different levels of noise are introduced to the image space. We observe the variance of the depth distribution.



(a) (e)

(b) (f)

(c) (g)

(d) (h)

Fig. 7. The variance σ_z of the depth Z within 5 frames and as a function of the noise amount in the image, and the depth is measured in cm. The amount of noise introduced in the image is 5 pixels in (a, b, c, d), and 15 pixels in (e, f, g, h). The norm of the depth error in the first row, the variance is in the second row, the image trajectory is in the third, while the pixel error is in the last row. The target positions of the features in the image space are marked by +.

Plot of the variance in addition to the estimated Gaussian densities, image space trajectories, features error are drawn in Fig 7.

The estimated distribution (mean value and variance) in presence of a 5 pixels and 15 pixels image noise are shown in Fig 7, (a and b) and (e and f) respectively. It is clear that the convergence time of the depth distribution is proportional to the amount of the noise. This could cause a local minima. In general, it can almost affect the convergence time and the trajectories in the image and Cartesian spaces, but it finally converge to its desired pose. This is depicted in Fig 7, (c, d), and (g, h) that shows the image trajectories (the desired feature positions are marked as +) and the feature error convergence.

4.2 Effect of the number of samples

The second experiment concerns with the number of samples used in the particle filter. The experiment is repeated four times with four different number of particles. Figure 8 depicts the effect of the number of samples used in the particle filter. It affects the final state of the estimated distribution variance. The final state value is slightly increased. The effect of the noise term that was introduced in the image is decreased by increasing the number of samples. It can be concluded that the robustness to image noise is directly related to the number of particles in the estimation process. It can be noted that the suitable number of samples for the depth estimation is 100 samples. This is a trade-off between the convergence speed and the steady state error of the depth estimate.

4.3 Evaluation of the depth value

The essential objective of this work is to estimate the depth value Z of the image features used in the image-based visual servoing control law. The probabilistic framework of the depth estimation gives multiple choices to the value of the depth Z substituted in the interaction matrix within the control law. Firstly, substituting the mean μ_z value of the probability density function of the depth estimation. The mean value converges very close to the real value in the second or third iteration. This gives a high accuracy in the evaluation of the depth value. The image trajectories, features error, velocity of the camera, and the norm of the difference vector between the real depth and the evaluated value are shown for this case in Fig. 9 (a, c, e, g). Secondly, after estimating the mean and variance of the distribution, generate a random sample of it and substitute in the control law. A larger variance final value is obtained by this method. The convergence time is a little longer. However, the second method looks more reasonable than the first one. The image trajectories, features error, velocity of the camera, and the norm of the difference vector between the real depth and the evaluated value are shown for this case in Fig. 9 (b, d, f, h).

We can conclude that the depth estimation of the features' depth is useful for visual servo control. It provides a kind of stability with respect to the depth estimates and the image noise. The convergence time of the estimation process can be affected directly by the image noise but it finally converges. Increasing the number of samples is not always useful, an optimal choice should be done to avoid large steady state error.

5. CONCLUSION AND FUTURE WORK

Image-based visual servoing is a simple and effective method comparing with other visual servoing method like position-based and hybrid visual servoing. The need to an estimation of the depth value of the image features used in the control law is the only weak point of it. It was proved that a rough estimation of the depth does not give a guarantee to the stability of the system, and the stability domain is not so wide. Estimating the depth distribution using particle filters gives a fine estimation of the depth. The simulation shows that the visual servoing system is stable even in presence of a considerable amount of

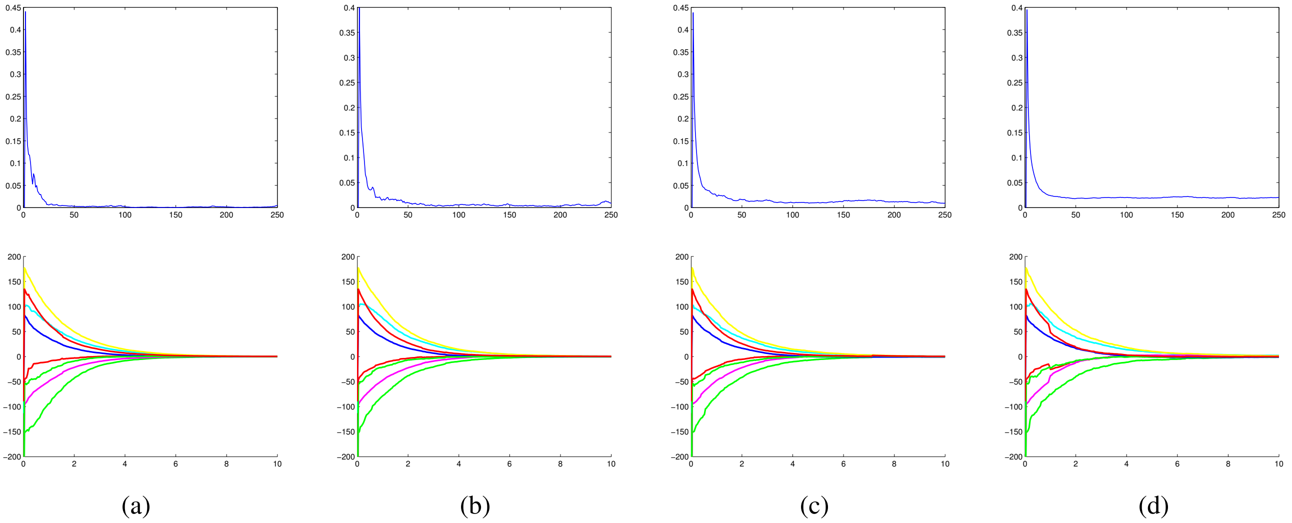


Fig. 8. This Figure shows the variance of the depth σ_z and as a function of the number of samples 50, 100, 1000, and 10000 samples in figures a, b, c, d respectively.

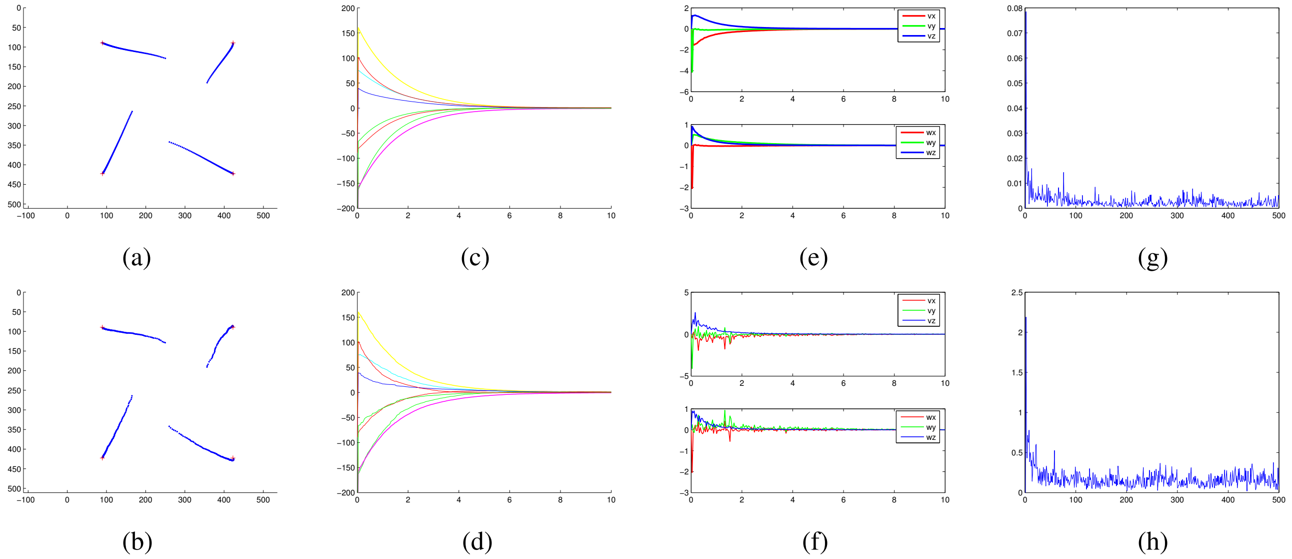


Fig. 9. This Figure shows the image trajectory, features error, screw velocity, and the norm of the difference between the estimated Z and the real value in (a, c, e, g) respectively, depth is measured in cm and velocity in m/sec and rad/sec. Using the time estimated mean as a depth value in the control law and generating random sample of estimated distribution in (b, d, f, h).

image noise. Hundred samples are more than enough to get a good estimation of the depth in visual servoing. This gives the applicability of the algorithm in the real-time application of visual servoing. The fine estimation of the depth increases the stability domain of the system with respect to the error in the depth estimation.

In future, we plan to merge the 3D estimation, represented by the depth of the image features, with the 2D information available in the image in such a way to improve and overcome the disadvantages of image-based visual servoing like the camera trajectory in the Cartesian space.

ACKNOWLEDGMENT

REFERENCES

Abdul Hafez, A.H. (2014). Visual servo control by optimizing hybrid objective function with visibility and path constraints. *Journal of Control Engineering and Applied Informatics*, 16(2), 120–129.

Abdul Hafez, A.H. and Cervera, E. (2014). Particle-filter-based pose estimation from controlled motion with application to visual servoing. *Int. Journal Adv. Robot Syst.*, 11, 1–11.

Bolic, M. (2004). *Architectures for Efficient Implementation of Particle Filters*. Ph.D. thesis, State University of New York, Stony Brook, USA.

Borangiu, T. (2004). *Intelligent Image Processing in Robotics and Manufacturing*. Romanian Academy Publishing House, Bucharest, first edition.

Chaumette, F. and Hutchinson, S. (2006). Visual servo control, part i: Basic approaches. *IEEE Robotics and Automation Magazine*, 13(4), 82–90.

Chen, J., Dixon, W.E., Dawson, D.M., and McIntyre, M.L. (2006). Homography-based visual servo tracking control of a wheeled mobile robot. *IEEE Transactions on Robotics*, 22(2), 406–415.

Collewet, C. and Chaumette, F. (2008). Visual servoing based on structure from controlled motion or on robust statistics. *IEEE Trans. on Robotics*, 24(2), 318–330.

Corke, P. and Hutchinson, S. (2001). A new partitioned approach to image -based visual servoing. *IEEE Transactions on Robotics and Automation*, 14(4), 507–515.

Corke, P.I. (2011). *Robotics, Vision & Control: Fundamental Algorithms in Matlab*. Springer.

Davison, A.J. (2003). Adaptive real-time particle filters for robot localization. In *Int. Conference on Computer Vision, ICCV'03*. France.

Deguchi, K. (1998). Optimal motion control for image-based visual servoingby decoupling translation and rotation. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, IROS'98*, volume 2, 705–711.

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

Hartley, R. and Zisserman, A. (2003). *Multiple View Geometry in Computer Vision*. Cambridge University Press, second edition.

Hutchinson, S., Hager, G., and Corke, P. (1996). A tutorial on visual servo control. *IEEE Transactions on Robotics and Automation*, 17, 18–27.

Kwok, C., Fox, D., and Meila, M. (2003). Adaptive real-time particle filters for robot localization. In *IEEE Int. Conference on Robotics and Automation, ICRA'03*.

Luca, A.D., Oriolo, G., and Giordano, P.R. (2007). On-line estimation of feature depth for image-based visual servoing schemes. In *IEEE Int. Conf. on Robotics and Automation, ICRA'07*, 2823–2828. IEEE.

Malis, E. and Chaumette, F. (2002). 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.

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

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

Mao, S., Huang, X., and Wang, M. (2012). Image jacobian matrix estimation based on online support vector regression. *Int. Journal Adv. Robot Syst.*, 9, 1–11.

Namiki, A., Hashimoto, K., and Ishikawa, M. (2003). A heirachical control architecture for high-speed visual servoing. *Int. Journal of Robotics Research, IJRR'03*, 22(10-11), 8873–888.

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

Robuffo Giordano, P., Spica, R., and Chaumette, F. (2014). An active strategy for plane detection and estimation with a monocular camera. In *IEEE Int. Conf. on Robotics and Automation, ICRA'14*, 4755–4761. Hong Kong, China.

Sanderson, A. and Weiss, L. (1980). Image-based visual servo control using relational graph error signal. *Proceeding IEEE*, 1074–1077.

Shirai, Y. and Inoue, H. (1973). Guiding a robot by visual feedback in assembling tasks. *Pattern Recognition*, 5, 99–108.