Measurement Errors in Visual Servoing

V. Kyrki *,[1]

Laboratory of Information Processing, Lappeenranta University of Technology, P.O. Box 20, 53851 Lappeenranta, Finland

D. Kragic, H.I. Christensen

Centre of Autonomous Systems, Royal Institute of Technology, Sweden

Abstract

This paper addresses the issue of measurement errors in visual servoing. The error characteristics of the vision based state estimation and the associated uncertainty of the control are investigated. The major contribution is the analysis of the propagation of image error through pose estimation and visual servoing control law. Using the analysis, two classical visual servoing methods are evaluated: position-based and 2 1/2 D visual servoing. The evaluation offers a tool to build and analyze hybrid control systems such as switching or partitioning control.

Key words: visual servoing, measurement errors, error estimation

1 Introduction

The use of visual feedback for closed loop control of a robot motion termed *visual servoing* has received a significant amount of attention during the last two decades [1], [2]. Most of the key research problem have been related to the performance of visual servoing methods in the presence of measurement and system modeling errors. As a result, a number of variations and hybrid visual servoing approaches have been proposed in the literature to cope with the inherit problems of image and position based visual servoing.

* Corresponding author.
Email addresses: kyrki@lut.fi (V. Kyrki), danik@nada.kth.se (D. Kragic), hic@nada.kth.se (H.I. Christensen).

[1] V. Kyrki was supported by a grant from the Academy of Finland.

Specific problems such as the effect of camera calibration errors have been studied in [3]. The convergence properties of the control part of the systems are known for most cases as discussed in [4,5]. While the convergence of the system is an essential performance property, it does not reveal much about the generated robot trajectory and its uncertainty.

The procedures of camera calibration have improved enormously over the last decade. However, even perfect calibration does not overcome the restriction of the image resolution and the imaging process causes an uncertainty in the control. Motivated by this fact, in this work we proposes the use of error propagation in the analysis and comparison of different types of visual servoing methods, i.e., position-based [1] and hybrid [6].

To provide a common ground for the modeling process, we start with a basic notation for modeling of a visual servoing system considered in this work as shown in Figure 1. Here, s denotes image measurements, and ${}^{o}T_{c}^{*}$ denotes the rigid body transformation relating the desired camera pose (position and orientation) to a measured target object. Similarly, ${}^{c}T_{o}$ is the current measured pose of the object w.r.t. the camera. The system is divided into three parts: pose estimation, servoing strategy, and control strategy. This model can be used with most position-based and hybrid approaches. It is based on eye-in-hand configuration, and the objective of servoing is defined as bringing the camera to a desired pose with respect to the target which is commonly termed *teach-by-showing* approach. The pose estimation part may compute the full 3-D pose of the target, or it may use homography- or epipolar-based techniques to infer partial pose. In the case of hybrid approaches, image features are directly used to control specific degrees of freedom. The choice of servoing strategy is based on the modeling of an error function and thus has a major effect on the robot's trajectory. On the other hand, the control strategy, such as for example a proportional control law, affects convergence properties especially in the case of a moving target. The related coordinate systems are illustrated in Figure 2.

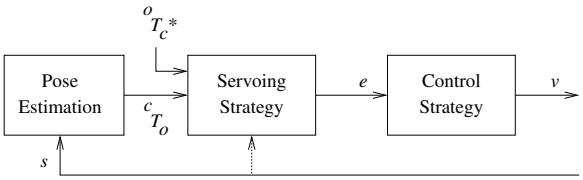

Fig. 1. System model.

To compare the visual servoing approaches with respect to image error modeling methods, a common reference has to be defined. We use the control output of a Cartesian controller as the reference, since it seems reasonable to study the sensitivity of the system by propagating the errors in the image measurements to the actual actuator motion. Thus, our approach predicts the uncertainty of

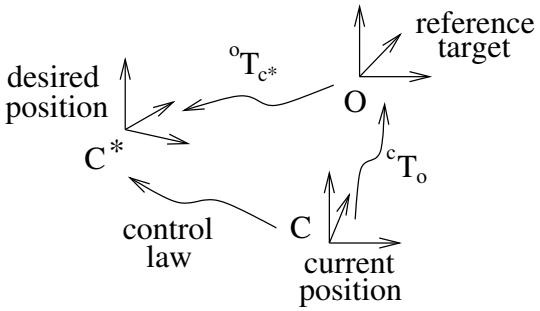

Fig. 2. Coordinate systems.

the Cartesian control with respect to the uncertainty in the image measurements. We do not wish to analyze uncertainty in the image measurements in detail, but instead model the image uncertainty with a Gaussian distribution, since a wide variety of methods is applicable for the image plane tracking. In this paper, we do not consider the correspondence problem since, in visual servoing, the correspondences can be tracked one frame to frame basis thus simplifying the correspondence problem. The division of the servoing model into subsystems allows us not only to compare the behavior of complete systems but also to compare their components. Furthermore, the error analysis is used to compare position-based visual servoing [1] and the hybrid approach termed 2.5D visual servoing proposed in [6].

The rest of the paper is organized as follows: We survey related work in Section 2 and provide a motivation for the research presented here. In Section 3 we describe the pose estimation algorithm and analyze its error propagation. Sections 4 and 5 present the position-based and hybrid servoing and their analysis. Section 6 extends the analysis of pose estimation uncertainty to cover all optimization-based algorithms. The analytical results are verified by Monte-Carlo simulation experiments in Section 7, which also discusses the merits of the different approaches. Finally, in Section 8, we present a summary and conclusions. Parts of this work has been presented in [7].

2 Related work

The work presented in this paper is closely related to the analysis of pose estimation algorithms. The most common approach considers the case where a 3D model of the object is given and its pose is estimated based on 2D-3D correspondences. Although a widely studied problem, most of the proposed methods are iterative which is a disadvantage in the context of visual servoing where "real-time" is a requirement. There are a few closed form solutions for point feature based pose estimation using any number of feature points [8–11]. In this paper, we have used the algorithm proposed by Fiore [10] together with

methods proposed in [12] and [13]. It has to be noted that in the proposed evaluation framework any other pose estimation algorithm can be used.

Although there are analyzes of sensitivity based on linear error propagation (e.g. [13]) in structure-from-motion research, according to the authors knowledge no corresponding analyzes have been published for the pose estimation. Haralick has demonstrated empirically that pose estimation breaks down when the image noise exceeds a certain threshold [14]. Ansar et al. have presented a sensitivity analysis where the results are upper bounds for the error derived from matrix perturbation theory [11]. Their experiments reveal that the bounds are highly conservative and thus not well suited for comparing different systems which we deal with in this work.

The error characteristics of visual servoing are usually investigated from either of the following two points of view: the stability of the closed-loop system, or the steady-state error [15]. It is known that the convergence of position-based visual servoing (PBVS) might be inhibited by the loss of stability in pose estimation [4]. 2.5D servoing does not seem to suffer from this problem [5], unless the partial pose estimation becomes unstable. Deng [15] has proposed use of the steady-state error as a measure of sensitivity of visual servoing. However, if long trajectories are executed, it is important to estimate the sensitivity of the system along the trajectory to, for example, predict the set of adequate trajectories in the presence of errors. Another approach is to consider the outliers in the image data. Comport et al. [16] have proposed a scheme to increase the robustness by embedding the outlier processing into the control law. Outlier rejection can also be performed in the image processing step [17].

Recently, Gans et al. [18] have proposed switching between position- and image-based servoing. We foresee that our error modeling can be used in the design of switching strategies which is currently an unsolved problem.

3 Pose estimation

In this section, we first describe the pose estimation algorithm used. It is based on earlier work by Fiore [10] and Weng et al.[13]. This is followed by the analysis of error propagation. The correspondence problem is not treated here because in visual servoing it applies mostly only to the initial estimation of the pose, as the features can be tracked from frame to frame. Initially, the correspondences can be found for example using interest point descriptors, e.g. [19].

3.1 Estimation algorithm

The pose estimation, also known as the exterior orientation problem, seeks the similarity transform consisting of translation $\mathbf{t}$ and rotation $\mathbf{R}$ that brings a set of known 3D feature points $\mathbf{a}_i$ into alignment with a set of corresponding image plane projections (x_i, y_i). Without loss of generality, we can assume unit focal length of the camera. Then, translation and rotation are the ones that best satisfy the set of equations

$$l_i \begin{bmatrix} x_i & y_i & 1 \end{bmatrix}^T = s\mathbf{R}(\mathbf{a}_i + \mathbf{t}), \;\; i = 1, \ldots, N \tag{1}$$

where l_i are the projective parameters, s is a scale factor, and N is the number of feature points.

In this work, the parameters l_i are first solved following [10]. We seek linear combinations of the N equations in (1) such that the right hand side becomes zero. Defining the data matrix $\mathbf{P}$ for 3D points as

$$\mathbf{P} = \begin{pmatrix} \mathbf{a}_1 & \cdots & \mathbf{a}_N \\ 1 & \cdots & 1 \end{pmatrix} \tag{2}$$

we can find the linear combinations by finding the weight matrix $\mathbf{W} \in \mathbb{R}^{N \times (N-4)}$ which satisfies

$$\mathbf{PW} = \mathbf{0}. \tag{3}$$

Thus, $\mathbf{W}$ represents the null space of $\mathbf{P}$ and it can be found from the singular value decomposition (SVD) of $\mathbf{P}$ as the matrix of the $N - 4$ right singular vectors of $\mathbf{P}$ corresponding to the null space. Because of this, for each of the linear combinations, each corresponding to a column of $\mathbf{W}$, we have

$$\sum_{i=1}^{N} w_{ij} l_i \begin{bmatrix} x_i \\ y_i \\ 1 \end{bmatrix} = \mathbf{0} \quad \text{for } j = 1, \ldots, N-4. \tag{4}$$

Defining the vector of projective parameters $\mathbf{l} = [l_1, \ldots, l_N]$, we can write this as $\mathbf{Gl} = \mathbf{0}$ where

$$\mathbf{G} = \begin{bmatrix} w_{1,1} \begin{bmatrix} x_1 & y_1 & 1 \end{bmatrix}^T & \cdots & w_{N,1} \begin{bmatrix} x_N & y_N & 1 \end{bmatrix}^T \\ \vdots & & \vdots \\ w_{1,1} \begin{bmatrix} x_1 & y_1 & 1 \end{bmatrix}^T & \cdots & w_{N,1} \begin{bmatrix} x_N & y_N & 1 \end{bmatrix}^T \end{bmatrix}. \tag{5}$$

Noting that every third row of $\mathbf{G}$ is a column of $\mathbf{W}$, $\mathbf{l}$ must be in the left null space of $\mathbf{W}$, spanned by $\mathbf{P}^T$. Therefore, we can write $\mathbf{l} = \mathbf{P}^T\boldsymbol{\alpha}$ for an

unknown vector $\boldsymbol{\alpha} \in \mathbb{R}^4$. Then, $\boldsymbol{\alpha}$ can be found as the solution to the set of homogeneous linear equations

$$\mathbf{C}\boldsymbol{\alpha} \equiv \begin{bmatrix} W_{1,1}\binom{x_1}{y_1} & \cdots & W_{N,1}\binom{x_N}{y_N} \\ \vdots & & \vdots \\ W_{1,N-4}\binom{x_1}{y_1} & \cdots & W_{N,N-4}\binom{x_N}{y_N} \end{bmatrix} \mathbf{P}^T\boldsymbol{\alpha} = \mathbf{0}. \tag{6}$$

Usually the system is overconstrained, and the solution can be found as the eigenvector corresponding to the smallest eigenvalue of $\mathbf{C}^T\mathbf{C}$. This in turn gives the set of projective parameters $\mathbf{l}$.

Now, we only need to recover the absolute orientation with scaling. With l_i known, we can write (1) as

$$\mathbf{b}_i = s\mathbf{R}(\mathbf{a}_i + \mathbf{t}), \;\; i = 1, \ldots, N \tag{7}$$

where $\mathbf{b}_i = [l_i x_i, l_i y_i, l_i]^T$. The unknown scale parameter s can be solved by centering the two point sets $\mathbf{a}_i$ and $\mathbf{b}_i$, and inspecting the ratio between the lengths of the centered vectors. Let $\bar{\mathbf{a}}$ and $\bar{\mathbf{b}}$ denote the means of sets $\mathbf{a}_i$ and $\mathbf{b}_i$. Then, the centered vectors can be defined as $\tilde{\mathbf{a}}_i = \mathbf{a}_i - \bar{\mathbf{a}}$ and $\tilde{\mathbf{b}}_i = \mathbf{b}_i - \bar{\mathbf{b}}$. Now, as the lengths of the centred point vectors are invariant to rotation, we can ignore the rotation and find the optimal least-squares scale from

$$s = \frac{\sum_i \|\tilde{\mathbf{a}}_i\| \|\tilde{\mathbf{b}}_i\|}{\sum_i \|\tilde{\mathbf{a}}_i\|^2}. \tag{8}$$

With s known, we want to find the rotation matrix that minimizes the sum of the square errors between the centered point sets, that is, $\sum_i \|\tilde{\mathbf{b}}_i - s\mathbf{R}\tilde{\mathbf{a}}_i\|_2^2$, which can also be written in matrix form as

$$\|\mathbf{B} - \mathbf{R}\mathbf{A}\|_F^2 \tag{9}$$

where

$$\mathbf{A} = s\begin{bmatrix} \tilde{\mathbf{a}}_1 & \ldots & \tilde{\mathbf{a}}_N \end{bmatrix} \qquad \mathbf{B} = \begin{bmatrix} \tilde{\mathbf{b}}_1 & \ldots & \tilde{\mathbf{b}}_N \end{bmatrix}$$

and $\|\cdot\|_F$ denotes the Frobenius norm. This, so called Orthogonal Procrustes problem, can be solved using SVD as suggested by Fiore, but we choose to solve the rotation using unit quaternions as presented in [13], as they have already proposed a suitable error analysis. The method was actually proposed already in [12]. We next give an outline of the method, and ask the reader to consult details in the original source.

A rotation can be represented with a unit quaternion $\mathbf{q}$ such that for any vector $\mathbf{a}$

$$\mathbf{q} * \mathbf{a} * \tilde{\mathbf{q}} \hat{=} \mathbf{R}\mathbf{a} \tag{10}$$

where $\tilde{\mathbf{q}}$ is the quaternion conjugate of $\mathbf{q}$, $*$ is the quaternion product and $\mathbf{R}$ is the corresponding rotation matrix. The quaternion product used in (10) can be written as a multiplication by a matrix, similarly to writing the cross product of vectors using a skew-symmetric matrix. After some manipulation using the basic properties of quaternions, the minimization problem (9) can be written using the matrix representation of quaternions as minimization of

$$\min_{\|q\|=1} \mathbf{q}^T \mathbf{E} \mathbf{q} \tag{11}$$

where $\mathbf{E}$ is defined as

$$\mathbf{E} = \sum_{i=1}^{N} \mathbf{E}_i^T \mathbf{E}_i \quad \text{with} \quad \mathbf{E}_i = \begin{pmatrix} 0 & (\mathbf{A}_i - \mathbf{B}_i)^T \\ \mathbf{B}_i - \mathbf{A}_i & [\mathbf{A}_i + \mathbf{B}_i]_\times \end{pmatrix}. \tag{12}$$

Here $\mathbf{A}_i$ and $\mathbf{B}_i$ denote the ith column of $\mathbf{A}$ and $\mathbf{B}$, respectively, and $[\cdot]_\times$ denotes the skew-symmetric matrix corresponding to cross product. The unit quaternion $\mathbf{q}$ representing the rotation can be found by minimization of (11) as the eigenvector corresponding to the smallest eigenvalue of $\mathbf{E}$.

When the quaternion $\mathbf{q} = (q_0, q_1, q_2, q_3)^T$ is known, the rotation matrix can be calculated as

$$\mathbf{R} = \begin{pmatrix} q_0^2+q_1^2-q_2^2-q_3^2 & 2(q_1q_2-q_0q_3) & 2(q_1q_3+q_0q_2) \\ 2(q_1q_2+q_0q_3) & q_0^2-q_1^2+q_2^2-q_3^2 & 2(q_2q_3-q_0q_1) \\ 2(q_1q_3-q_0q_2) & 2(q_0q_1+q_2q_3) & q_0^2-q_1^2-q_2^2+q_3^2 \end{pmatrix}. \tag{13}$$

Finally, the translation is found from $\mathbf{t} = s^{-1}\mathbf{R}^T\bar{\mathbf{b}} - \bar{\mathbf{a}}$ where $\bar{\mathbf{a}}$ and $\bar{\mathbf{b}}$ are the point set centroids.

3.2 Error analysis

The error analysis in this paper is based on first-order error propagation [20]. The goal of this analysis is to determine the covariance of the pose estimate with respect to the variances of image plane coordinates. While errors can also be analyzed by finding worst case error bounds, this can result in overly conservative bounds that are suitable only for small errors. In practice, the possible redundancy of data in pose estimation (i.e., having more features than necessary) allows finding stable solutions also in the presence of noise. In this paper, it is assumed that the errors in the pose estimate result from the noise in the image coordinates of features. Their sources include spatial quantization, feature detection, and camera distortion. However, we assume that there is no systematic calibration error and thus the image noise can be modeled as zero-mean random variables. It is further assumed that the errors between points are uncorrelated.

Let $\mathbf{x}$ be the vector of image coordinates of features such that

$$\mathbf{x} = (x_1, \ldots, x_N, y_1, \ldots, y_N)^T.$$

We formulate the error analysis problem as finding the matrices $\mathbf{D_t}$ and $\mathbf{D_R}$ such that $\boldsymbol{\delta}_\mathbf{t} = \mathbf{D_t}\boldsymbol{\delta}_\mathbf{x}$ and $\boldsymbol{\delta}_\mathbf{R} = \mathbf{D_R}\boldsymbol{\delta}_\mathbf{x}$ are linear error estimates in $\mathbf{t}$ and $\mathbf{R}$ with respect to errors in $\mathbf{x}$. In other words, the measurements $\mathbf{x}$ are considered to be affected by additive noise such that $\mathbf{x} = \mathbf{x}_0 + \boldsymbol{\delta}_\mathbf{x}$ where $\mathbf{x}_0$ is the true value. We have then a function $\mathbf{t}(\mathbf{x})$ for which we inspect its behavior around the current measurements using a first-order Taylor expansion. This allows us to write the estimate of the error in $\mathbf{t}(\mathbf{x})$ as $\boldsymbol{\delta}\mathbf{t}(\mathbf{x}) = \frac{\partial \mathbf{t}(\mathbf{x})}{\partial \mathbf{x}}\boldsymbol{\delta}\mathbf{x}$. Here, $\mathbf{D_t}$ is just the gradient $\frac{\partial \mathbf{t}(\mathbf{x})}{\partial \mathbf{x}}$. Going through several steps of the algorithm, the chain-rule of differentiation can be used in determining the gradient. That is, the gradient for each step can be calculated separately.

For vectors, let $\boldsymbol{\Gamma}$ denote the covariance matrix, e.g., $\boldsymbol{\Gamma}_\mathbf{x} = E[\boldsymbol{\delta}_\mathbf{x}\boldsymbol{\delta}_\mathbf{x}{}^T]$. If the covariance of the input and the linear linear mappings are known, the covariance matrices of the rotation and translation can be written $\boldsymbol{\Gamma}_\mathbf{R} = \mathbf{D_R}\boldsymbol{\Gamma}_\mathbf{x}\mathbf{D_R}{}^T$ and $\boldsymbol{\Gamma}_\mathbf{t} = \mathbf{D_t}\boldsymbol{\Gamma}_\mathbf{x}\mathbf{D_t}{}^T$. Note that matrix $\mathbf{R}$ must be represented as a vector $\mathbf{r}$ by concatenating the columns of the matrix into a single vector. Thus, $\boldsymbol{\delta}_\mathbf{R}$ is the error in this vector. This notation will be adopted for denoting vectors corresponding to matrices such that $\mathbf{c}$ is a vector corresponding to matrix $\mathbf{C}$.

It is evident that $\mathbf{D_t}$ and $\mathbf{D_R}$ depend on the values of both $\mathbf{x}$ and every $\mathbf{a}_i$, that is, the image measurements and the object model. The error is now propagated through the pose estimation algorithm. First, it can be seen that $\mathbf{W}$ in (3) depends only on matrix $\mathbf{P}$ where there is no associated uncertainty. The uncertainty in matrix $\mathbf{C}$ (6) can be found by finding the matrix $\mathbf{G_C}$ that represents the transform from $\mathbf{x}$ to $\mathbf{c}$ (and $\boldsymbol{\delta}_\mathbf{c} = \mathbf{G_C}\boldsymbol{\delta}_\mathbf{x}$), the vector representation of matrix $\mathbf{C}$. This operation is linear so no approximations are needed. The matrix is easily found to be

$$\mathbf{G_C} = \begin{pmatrix} \mathbf{Q_1} & \mathbf{0} & \mathbf{Q_2} & \mathbf{0} & \mathbf{Q_3} & \mathbf{0} & \mathbf{Q_4} & \mathbf{0} \\ \mathbf{0} & \mathbf{Q_1} & \mathbf{0} & \mathbf{Q_2} & \mathbf{0} & \mathbf{Q_3} & \mathbf{0} & \mathbf{Q_4} \end{pmatrix}^T \tag{14}$$

where

$$\mathbf{Q}_i = \begin{pmatrix} W_{1,1}P_{i,1} & \cdots & W_{1,N-4}P_{i,1} \\ \vdots & & \vdots \\ W_{N,1}P_{i,N} & \cdots & W_{N,N-4}P_{i,N} \end{pmatrix}.$$

Next, the linear estimate for the error in $\mathbf{C}^T\mathbf{C}$ is found. Denoting the error matrix corresponding to vector $\boldsymbol{\delta}_\mathbf{c}$ by $\boldsymbol{\Delta}_\mathbf{C}$, the linear estimate is

$$\boldsymbol{\Delta}_{\mathbf{C}^T\mathbf{C}} \approx \mathbf{C}^T\boldsymbol{\Delta}_\mathbf{C} + \boldsymbol{\Delta}_\mathbf{C}{}^T\mathbf{C}. \tag{15}$$

Using the vector notation, this can be written as

$$\boldsymbol{\delta}_{\mathbf{C}^T\mathbf{C}} \approx \mathbf{G}_{\mathbf{C}^T\mathbf{C}}\boldsymbol{\delta}_\mathbf{c} = \mathbf{G}_{\mathbf{C}^T\mathbf{C}}\mathbf{G_C}\boldsymbol{\delta}_\mathbf{x} = \mathbf{D}_{\mathbf{C}^T\mathbf{C}}\boldsymbol{\delta}_\mathbf{x} \tag{16}$$

where $\mathbf{G}_{\mathbf{C}^T\mathbf{C}} = [\mathbf{K}_{ij}] + [\mathbf{L}_{ij}]$ can be determined using (15). $[\mathbf{K}_{ij}]$ and $[\mathbf{L}_{ij}]$ are matrices with $4 \times 2(N-4)$ submatrices $\mathbf{K}_{ij}$ and $\mathbf{L}_{ij}$, with $\mathbf{K}_{ij}$ equal to $\mathbf{C}^T$ when $i = j$ and otherwise it is a zero matrix. In $\mathbf{L}_{ij}$, the jth row is equal to the ith row of $\mathbf{C}^T$, otherwise the elements are zero.

To propagate the error through the eigenvalue decomposition, we use the result presented by Weng et al. in [13]. The linear error term in $\boldsymbol{\alpha}$, the smallest eigenvector of $\mathbf{C}^T\boldsymbol{C}$, is given by

$$\begin{aligned}\boldsymbol{\delta}_{\boldsymbol{\alpha}} &\approx \mathbf{H}\boldsymbol{\Delta}\mathbf{H}^T\boldsymbol{\Delta}_{\mathbf{C}^T\mathbf{C}}\boldsymbol{\alpha} \\ &= \mathbf{H}\boldsymbol{\Delta}\mathbf{H}^T[\alpha_1 I_4 \quad \alpha_2 I_4 \quad \alpha_3 I_4 \quad \alpha_4 I_4]\boldsymbol{\delta}_{\mathbf{C}^T\mathbf{C}} \\ &= \mathbf{G}_{\boldsymbol{\alpha}}\boldsymbol{\delta}_{\mathbf{C}^T\mathbf{C}} = \mathbf{G}_{\boldsymbol{\alpha}}\mathbf{D}_{\mathbf{C}^T\mathbf{C}}\boldsymbol{\delta}_{\mathbf{x}} = \mathbf{D}_{\boldsymbol{\alpha}}\boldsymbol{\delta}_{\mathbf{x}}\end{aligned} \tag{17}$$

where $\mathbf{H}$ is the matrix of eigenvectors of $\mathbf{C}^\mathbf{T}\mathbf{C}$ and $\boldsymbol{\Delta}$ is given in terms of the eigenvalues λ_i as

$$\boldsymbol{\Delta} = \operatorname{diag}\left\{0, (\lambda_1 - \lambda_2)^{-1}, (\lambda_1 - \lambda_3)^{-1}, (\lambda_1 - \lambda_4)^{-1}\right\}.$$

For a proof, see Appendix A of [13].

As the projective parameters depend linearly on $\boldsymbol{\alpha}$, we can find the associated error as $\boldsymbol{\delta}_\mathbf{l} = \mathbf{P}^T\boldsymbol{\delta}_{\boldsymbol{\alpha}} \approx \mathbf{P}^T\mathbf{D}_{\boldsymbol{\alpha}}\boldsymbol{\delta}_{\mathbf{x}} = \mathbf{D}_\mathbf{l}\boldsymbol{\delta}_{\mathbf{x}}$.

We now continue to propagate the errors to $\mathbf{b}_i$. Let

$$\boldsymbol{\delta}_\mathbf{B} = [\delta_{l_1x_1}, \cdots, \delta_{l_Nx_N}, \delta_{l_1y_1}, \cdots, \delta_{l_Ny_N}, \delta_{l_1}, \cdots, \delta_{l_N}].$$

The linear approximation for the error is

$$\begin{aligned}\boldsymbol{\delta}_\mathbf{B} &\approx \begin{pmatrix} \operatorname{diag}(\mathbf{l}) & \mathbf{0} & \operatorname{diag}(\mathbf{x_{1\ldots N}}) \\ \mathbf{0} & \operatorname{diag}(\mathbf{l}) & \operatorname{diag}(\mathbf{y_{1\ldots N}}) \\ \mathbf{0} & \mathbf{0} & \mathbf{I} \end{pmatrix} \begin{pmatrix} \boldsymbol{\delta}_\mathbf{x} \\ \boldsymbol{\delta}_\mathbf{l} \end{pmatrix} \\ &= \begin{pmatrix} \operatorname{diag}(\mathbf{l})+\operatorname{diag}(\mathbf{x_{1\ldots N}})\mathbf{D_l} & \mathbf{0} \\ \mathbf{0} & \operatorname{diag}(\mathbf{l})+\operatorname{diag}(\mathbf{y_{1\ldots N}})\mathbf{D_l} \\ \multicolumn{2}{c}{\mathbf{D_l}} \end{pmatrix} \boldsymbol{\delta}_\mathbf{x} \\ &= \mathbf{D}_\mathbf{B}\boldsymbol{\delta}_\mathbf{x}\end{aligned} \tag{18}$$

In the following, we will skip the details on linear steps of the error propagation to keep the discussion as brief as possible while still stating each approximation during the nonlinear steps. Centering the set of vectors $\mathbf{b}_i$ does not involve nonlinear operations so no approximations need to be done to find the error in $\tilde{\mathbf{b}}_i$. Then, $\boldsymbol{\delta}_{\tilde{\mathbf{B}}} \approx \mathbf{D}_{\tilde{\mathbf{B}}}\boldsymbol{\delta}_\mathbf{x}$. In calculating the scale, $\delta_s \approx \mathbf{G}_\mathbf{s}\boldsymbol{\delta}_{\tilde{\mathbf{B}}}$ where

$$\mathbf{G}_\mathbf{s} = \frac{1}{\sum_i \|\tilde{\mathbf{a}}_i\|^2}\left(\frac{\tilde{b}_{1,1}\|\mathbf{a}_1\|}{\|\tilde{\mathbf{b}}_1\|} \ldots \frac{\tilde{b}_{N,1}\|\mathbf{a}_N\|}{\|\tilde{\mathbf{b}}_N\|}\right) \tag{19}$$

As stated before, the rotation matrix is now estimated using unit quaternions. This encompasses another case of determining the eigenvector corresponding

to the smallest eigenvalue of a matrix $\mathbf{E}$, which is a non-linear combination of previously known variables. Its error $\boldsymbol{\delta}_\mathbf{E}$ can be found by first finding the errors in $\mathbf{E}_i$ which are linear with respect to errors in $\mathbf{B}$. The error in the matrix multiplication $\mathbf{E}_i^T\mathbf{E}_i$ can be propagated as in (15). Finally, the error in $\mathbf{E}$ can be approximated as a matrix product $\boldsymbol{\delta}_\mathbf{E} = \mathbf{G}_\mathbf{E}[\boldsymbol{\delta}_\mathbf{B}^T, \delta_s]^T$.

The error can now be propagated in a similar fashion as shown above for vector $\boldsymbol{\alpha}$ in (17). As a result, we get the unit quaternion $\mathbf{q}$ that represents the rotation and its error with respect to the errors in input $\boldsymbol{\delta}_\mathbf{q} \approx \mathbf{G}_\mathbf{q}\boldsymbol{\delta}_\mathbf{E} = \mathbf{D}_\mathbf{q}\boldsymbol{\delta}_\mathbf{x}$. We can estimate the first order perturbation of $\mathbf{R}$ as $\boldsymbol{\delta}_\mathbf{R} \approx \mathbf{G}_\mathbf{R}\boldsymbol{\delta}_\mathbf{q} = \mathbf{D}_\mathbf{R}\boldsymbol{\delta}_\mathbf{x}$ where

$$\mathbf{G}_\mathbf{R} = 2\begin{pmatrix} q_0 & q_3 & -q_2 & -q_3 & q_0 & q_1 & q_3 & -q_1 & q_0 \\ q_1 & q_2 & q_3 & q_2 & -q_1 & q_0 & q_3 & -q_0 & -q_1 \\ -q_2 & q_1 & -q_0 & q_1 & q_2 & q_3 & q_0 & q_3 & -q_2 \\ -q_3 & q_0 & q_1 & -q_0 & -q_3 & q_2 & q_1 & q_2 & q_3 \end{pmatrix}^T \tag{20}$$

which results directly from the differentiation of (13). The error in the translation can finally be estimated from the first-order Taylor expansion as

$$\begin{aligned} \boldsymbol{\delta}_\mathbf{t} &\approx \frac{1}{s}(\mathbf{R}^T\boldsymbol{\delta}_{\bar{\mathbf{b}}} + \boldsymbol{\Delta}_\mathbf{R}{}^T\bar{\mathbf{b}}) - \frac{1}{s^2}\delta_s\mathbf{R}^T\bar{\mathbf{b}} \\ &= \mathbf{G}_\mathbf{t}\boldsymbol{\delta}_\mathbf{E} = \mathbf{D}_\mathbf{t}\boldsymbol{\delta}_\mathbf{x} \end{aligned} \tag{21}$$

In summary, we have expressed the perturbations in the pose estimate as a linear transformation of the perturbations in the input image. This allows us also to write the covariance matrices of the pose parameters as

$$\boldsymbol{\Gamma}_\mathbf{R} = \mathbf{D}_\mathbf{R}\boldsymbol{\Gamma}_\mathbf{x}\mathbf{D}_\mathbf{R}{}^T \qquad \boldsymbol{\Gamma}_\mathbf{t} = \mathbf{D}_\mathbf{t}\boldsymbol{\Gamma}_\mathbf{x}\mathbf{D}_\mathbf{t}{}^T. \tag{22}$$

The following two sections outline two visual servoing methods and relate the uncertainty in the pose estimate presented in this section to the uncertainty in the control.

4 Position Based Visual Servoing

In position-based visual servoing (PBVS), the task function is defined in terms of the pose transformation between the current and the desired position, which can be expressed as the transformation ${}^cT_{c^*}$ (see Figure 2). The input image is usually used to estimate the camera to object transformation ${}^cT_o$ which can be composed with the object to desired pose transformation ${}^oT_{c^*}$ to find the relation from the current to the desired pose. By decomposing the transformation

matrices into translation and rotation, this can be expressed as

$$\begin{aligned} {}^{c}T_{c^*} &= {}^{c}T_{o}\,{}^{o}T_{c^*} = \begin{pmatrix} {}^{c}\mathbf{R}_{o} & {}^{c}\mathbf{t}_{o} \\ \mathbf{0} & 1 \end{pmatrix} \begin{pmatrix} {}^{o}\mathbf{R}_{c^*} & {}^{o}\mathbf{t}_{c^*} \\ \mathbf{0} & 1 \end{pmatrix} \\ &= \begin{pmatrix} {}^{c}\mathbf{R}_{o}\,{}^{o}\mathbf{R}_{c^*} & {}^{c}\mathbf{R}_{o}\,{}^{o}\mathbf{t}_{c^*} + {}^{c}\mathbf{t}_{o} \\ \mathbf{0} & 1 \end{pmatrix} = \begin{pmatrix} {}^{c}\mathbf{R}_{c^*} & {}^{c}\mathbf{t}_{c^*} \\ \mathbf{0} & 1 \end{pmatrix} \end{aligned} \tag{23}$$

The task function for position is then the vector ${}^{c}\mathbf{t}_{c^*}$. For orientation, the rotation matrix can be decomposed into axis of rotation $\mathbf{u} = (u_1, u_2, u_3)^T$ and angle θ, which can be multiplied to attain the task function $\mathbf{u}\theta$ using

$$\begin{aligned} \theta &= \operatorname{acos}\left(\frac{\operatorname{trace}(\mathbf{R}) - 1}{2}\right) \\ \sin\theta &= \sqrt{1 - \left(\frac{\operatorname{trace}(\mathbf{R}) - 1}{2}\right)^2} \\ u_1 = \frac{R_{32} - R_{23}}{2\sin\theta} \quad u_2 &= \frac{R_{13} - R_{31}}{2\sin\theta} \quad u_3 = \frac{R_{21} - R_{12}}{2\sin\theta}. \end{aligned} \tag{24}$$

Thus, the position-based controller can be written

$$\mathbf{v} = -\lambda \begin{pmatrix} {}^{c}\mathbf{t}_{c^*} \\ \mathbf{u}\theta \end{pmatrix}. \tag{25}$$

Starting from the result of the analysis of pose estimation, we first inspect the camera to object transformation. The rotation matrix $\mathbf{R}$ in the image formation model in (1) is the desired rotation from the camera to object frames ${}^{c}\mathbf{R}_{o}$. The reference frame for the translation is expressed with respect to the object rather than the camera. Thus, we need to rotate the translation vector to correspond to camera frame axes, and find the uncertainty for this rotated vector using the uncertainties in both the rotation matrix and the translation vector. The uncertainty can thus be expressed as

$$\boldsymbol{\delta}_{{}^{c}\mathbf{t}_{o}} \approx \mathbf{G}_{{}^{c}\mathbf{t}_{o}} \begin{pmatrix} \mathbf{D}_{\mathbf{R}} \\ \mathbf{D}_{\mathbf{t}} \end{pmatrix} \boldsymbol{\delta}_{\mathbf{x}} = \mathbf{D}_{{}^{c}\mathbf{t}_{o}} \boldsymbol{\delta}_{\mathbf{x}}. \tag{26}$$

Assuming that there is no uncertainty associated with the desired position, the error in the rotation from the current to desired pose can be approximated as $\boldsymbol{\Delta}_{{}^{c}\mathbf{R}_{c^*}} \approx \boldsymbol{\Delta}_{{}^{c}\mathbf{R}_{o}}\,{}^{o}\mathbf{R}_{c^*}$ which can be expressed as $\boldsymbol{\delta}_{{}^{c}\mathbf{R}_{c^*}} \approx \mathbf{G}_{{}^{c}\mathbf{R}_{c^*}} \boldsymbol{\delta}_{{}^{c}\mathbf{R}_{o}}$. For the translation, the corresponding errors can be written

$$\boldsymbol{\delta}_{{}^{c}\mathbf{t}_{c^*}} \approx \boldsymbol{\Delta}_{{}^{c}\mathbf{R}_{o}}\,{}^{o}\mathbf{t}_{c^*} + \boldsymbol{\delta}_{{}^{c}\mathbf{t}_{o}} = \mathbf{G}_{{}^{c}\mathbf{t}_{c^*}} \begin{pmatrix} \boldsymbol{\delta}_{{}^{c}\mathbf{R}_{o}} \\ \boldsymbol{\delta}_{{}^{c}\mathbf{t}_{o}} \end{pmatrix} = \mathbf{D}_{{}^{c}\mathbf{t}_{c^*}} \boldsymbol{\delta}_{\mathbf{x}}. \tag{27}$$

What remains is to transform the rotation matrix into a control vector for

rotation. We use the $\mathbf{u}\theta$ form and estimate the errors as

$$\boldsymbol{\delta}_{\mathbf{u}\theta} \approx \mathbf{G}_{\mathbf{u}\theta}\boldsymbol{\delta}_{{}^{c}\mathbf{R}_{c^*}} = \mathbf{D}_{\mathbf{u}\theta}\boldsymbol{\delta}_{\mathbf{x}}, \tag{28}$$

where by differentiation of the product $\mathbf{u}\theta$

$$\mathbf{G}_{\mathbf{u}\theta} = \mathbf{u}\delta_\theta + \theta\boldsymbol{\delta}_{\mathbf{u}} = \mathbf{u}\mathbf{G}_\theta\boldsymbol{\delta}_{{}^{c}\mathbf{R}_{c^*}} + \theta\mathbf{G}_{\mathbf{u}} \begin{pmatrix} \boldsymbol{\delta}_{{}^{c}\mathbf{R}_{c^*}} \\ \delta_\theta \end{pmatrix}.$$

By partial differentiation of θ and $\mathbf{u}$ in (24) with respect to elements of $\mathbf{R}$, we can write

$$\mathbf{G}_\theta = -\frac{1}{2\sin\theta}(1,0,0,0,1,0,0,0,1) \tag{29}$$

and

$$\mathbf{G}_{\mathbf{u}} = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & \frac{1}{2\sin\theta} & 0 & -\frac{1}{2\sin\theta} & 0 & -\frac{u_1}{\tan\theta} \\ 0 & 0 & -\frac{1}{2\sin\theta} & 0 & 0 & 0 & \frac{1}{2\sin\theta} & 0 & 0 & -\frac{u_2}{\tan\theta} \\ 0 & \frac{1}{2\sin\theta} & 0 & -\frac{1}{2\sin\theta} & 0 & 0 & 0 & 0 & 0 & -\frac{u_3}{\tan\theta} \end{pmatrix}. \tag{30}$$

Assuming that a proportional control is used, the error in the control vector $\mathbf{v}$ is finally estimated as

$$\boldsymbol{\delta}_{\mathbf{v}} = -\lambda \begin{pmatrix} \boldsymbol{\delta}_{{}^{c}\mathbf{t}_{c^*}} \\ \boldsymbol{\delta}_{\mathbf{u}\theta} \end{pmatrix} \approx \begin{pmatrix} -\lambda\mathbf{D}_{{}^{c}\mathbf{t}_{c^*}} \\ -\lambda\mathbf{D}_{\mathbf{u}\theta} \end{pmatrix} \delta_x = \mathbf{D}_{\mathbf{v}}\boldsymbol{\delta}_{\mathbf{x}}. \tag{31}$$

This allows us also to approximate the covariance matrix of the control error from $\boldsymbol{\Gamma}_{\mathbf{v}} = E[\boldsymbol{\delta}_{\mathbf{v}}\boldsymbol{\delta}_{\mathbf{v}}^T] \approx \mathbf{D}_{\mathbf{v}}\boldsymbol{\Gamma}_{\mathbf{x}}{\mathbf{D}_{\mathbf{v}}}^T$. The covariance matrix now represents the uncertainty in the control caused by the uncertainty in the image measurements, and can be used to examine the uncertainty in the different degrees of freedom of the control either separately or together.

5 Hybrid Visual Servoing

The hybrid visual servoing approach, called 2.5D servoing, was originally presented as a method suitable for avoiding the target leaving the field of view of the camera (a PBVS problem), and to perform servoing without a complete 3D model of the target [6]. It is based on partial pose estimation using a scaled Euclidean reconstruction with a homography decomposition. However, it can be also used with full pose estimation.

We now briefly present the 2.5D servoing with full pose estimation used in our work. The control scheme is based on controlling the orientation using the estimated 3-D rotation between the current and desired poses and driving the vector $\mathbf{u}\theta$ to zero just as in PBVS. The position in turn is controlled using a

single point feature that is driven towards its desired location in both image coordinates and depth. Thus, the visibility of the feature during the servoing sequence is guaranteed. The task vector can be defined as

$$\mathbf{e} = [x - x^*, y - y^*, \log(Z/Z^*), \theta\mathbf{u}^T]^T \tag{32}$$

where (x, y) is the position of the control point in the image, Z is its depth, and asterisks denote the desired values. The motion control law is then

$$\mathbf{v} = -\lambda \begin{pmatrix} \mathbf{L}_v^{-1} & -\mathbf{L}_v^{-1}\mathbf{L}_{v\omega} \\ \mathbf{0} & \mathbf{I} \end{pmatrix} \mathbf{e} \tag{33}$$

where

$$\mathbf{L}_v^{-1} = \begin{pmatrix} -Z & 0 & -xZ \\ 0 & -Z & -yZ \\ 0 & 0 & -Z \end{pmatrix} \tag{34}$$

and

$$\mathbf{L}_{v\omega} = \begin{pmatrix} xy & -(1+x^2) & y \\ 1+y^2 & -xy & -x \\ -y & x & 0 \end{pmatrix}. \tag{35}$$

In our framework (Figure 1), the rotation $\mathbf{u}\theta$ and the depth Z are calculated using the pose estimation while x and y result directly from image measurements. Z can thus be written as

$$[X, Y, Z]^T = {}^cT_o[\mathbf{a}^T, 1]^T. \tag{36}$$

The sensitivity for the rotation is identical to that presented in the previous section. However, we desire to estimate the error in the control vector to recognize correlations between the errors in different variables. The error in the depth can be approximated in terms of the errors on estimated rotation and translation as

$$\delta_Z = \mathbf{G}_Z[\mathbf{D_R}^T, \mathbf{D_t}^T]^T\boldsymbol{\delta}_\mathbf{x} \tag{37}$$

where $\mathbf{G}_Z$ can be determined from (36). The uncertainty in the control output $\mathbf{v}$ can be approximated as $\boldsymbol{\delta}_\mathbf{v} = \mathbf{G}_\mathbf{v}[\delta_x, \delta_y, \delta_Z, \boldsymbol{\delta}_\mathbf{u}^T, \delta_\theta]^T = \mathbf{D}_\mathbf{v}\boldsymbol{\delta}_\mathbf{x}$ where $\mathbf{G}_\mathbf{v}$ can be determined by differentiation of (33) as

$$\begin{aligned} \mathbf{G_v} &= -\lambda \begin{pmatrix} \mathbf{G_{v1}} & \mathbf{G_{v2}} \end{pmatrix} \\ \mathbf{G_{v1}} &= \begin{pmatrix} -Z - Z\log\frac{Z}{Z^*} & Zu_3\theta & -2x + x^* - x\log\frac{Z}{Z^*} - u_2\theta + yu_3\theta \\ -Zu_3\theta & -Z - Z\log\frac{Z}{Z^*} & -2y + y^* - y\log\frac{Z}{Z^*} + u_1\theta - xu_3\theta \\ Zu_2\theta & -Zu_1\theta & -\log\frac{Z}{Z^*} - 1 - yu_1\theta + xu_2\theta \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \\ \mathbf{G_{v2}} &= \begin{pmatrix} 0 & -Z\theta & yZ\theta & -Zu_2 + yZu_3 \\ Z\theta & 0 & -xZ\theta & Zu_1 - xZu_3 \\ -yZ\theta & xZ\theta & 0 & xZu_2 - yZu_1 \\ \theta & 0 & 0 & u_1 \\ 0 & \theta & 0 & u_2 \\ 0 & 0 & \theta & u_3 \end{pmatrix}. \end{aligned} \tag{38}$$

Then, the covariance of the control is approximately $\boldsymbol{\Gamma}_\mathbf{v} \approx \mathbf{D}_\mathbf{v}\boldsymbol{\Gamma}_\mathbf{x}{\mathbf{D}_\mathbf{v}}^T$.

6 Optimization based pose estimation

In this section, we show how to extend the results of the error estimation of pose estimation to cover all optimization based pose estimation algorithms. The analysis method does not need to take into account a particular optimization method, because it only estimates how the location of the minimum changes when the input of minimization is disturbed. The analysis is based on [20].

Optimization-based pose estimation can be defined as minimizing the image plane error

$$e = \sum_i \left[\left(x_i - \frac{[1,0,0]^T \mathbf{R}(\mathbf{a}_i + \mathbf{t})}{[0,0,1]^T \mathbf{R}(\mathbf{a}_i + \mathbf{t})} \right)^2 + \left(y_i - \frac{[0,1,0]^T \mathbf{R}(\mathbf{a}_i + \mathbf{t})}{[0,0,1]^T \mathbf{R}(\mathbf{a}_i + \mathbf{t})} \right)^2 \right]. \quad (39)$$

While the rotation matrix $\mathbf{R}$ has nine entries, it has only three degrees of freedom. In minimization, the rotation can be represented as a three-element vector $\mathbf{w} = \theta\mathbf{u}$, which encodes the rotation as an angle-axis-pair.

First-order error propagation can be used to inspect the effect of uncertainty in 2-D and 3-D point coordinates to the location of the minimum. Let $\mathbf{\Theta} = \{\mathbf{t}, \mathbf{w}\}$ denote the true pose parameters and $\mathbf{X} = (\mathbf{a}_1, x_1, y_1, \ldots, \mathbf{a}_N, x_N, y_N)^T$ denote the set of coordinates without errors. The gradient of the error function with respect to the pose parameters can now be written as

$$g(\mathbf{X}, \mathbf{\Theta}) = \frac{\partial e(\mathbf{X}, \mathbf{\Theta})}{\partial \mathbf{\Theta}}. \quad (40)$$

By denoting the measurements by $\hat{\mathbf{X}} = \mathbf{X} + \Delta\mathbf{X}$ and corresponding pose parameters by $\hat{\mathbf{\Theta}} = \mathbf{\Theta} + \Delta\mathbf{\Theta}$, we can write the first order Taylor approximation for the gradient at $(\mathbf{X}, \mathbf{\Theta})$ using the measurements as

$$g(\mathbf{X}, \mathbf{\Theta}) = g(\hat{\mathbf{X}}, \hat{\mathbf{\Theta}}) - \frac{\partial g^T(\hat{\mathbf{X}}, \hat{\mathbf{\Theta}})}{\partial \mathbf{X}} \Delta\mathbf{X} - \frac{\partial g^T(\hat{\mathbf{X}}, \hat{\mathbf{\Theta}})}{\partial \mathbf{\Theta}} \Delta\mathbf{\Theta}. \quad (41)$$

The gradient of the error $g(\cdot)$ must be zero at both $(\mathbf{X}, \mathbf{\Theta})$ and $(\hat{\mathbf{X}}, \hat{\mathbf{\Theta}})$, so to a first order approximation

$$\Delta\mathbf{\Theta} = -\left(\frac{\partial g^T(\hat{\mathbf{X}}, \hat{\mathbf{\Theta}})}{\partial \mathbf{\Theta}} \right)^{-1} \frac{\partial g^T(\hat{\mathbf{X}}, \hat{\mathbf{\Theta}})}{\partial \mathbf{X}} \Delta\mathbf{X} = \mathbf{D}_{\Delta\Theta} \Delta\mathbf{X}. \quad (42)$$

The difference of this to the analysis of Sec. 3.2 is that now in addition to the measurement uncertainty, also the uncertainty of the 3D model can be taken into account. The uncertainty in pose parameters can then be propagated through visual servoing laws as shown in Sections 3 and 4. The analytical

forms of the gradients can be calculated using (39) and (40) but they are omitted here for the sake of brevity.

Similarly to the closed-form pose estimation, we can estimate the covariance of $\mathbf{\Theta}$ by denoting the covariance matrix of $\mathbf{X}$ by $\mathbf{\Gamma_X}$ as

$$\mathbf{\Gamma_\Theta} = \mathbf{D}_{\Delta\Theta}\mathbf{\Gamma_X}\mathbf{D}_{\Delta\Theta}^T. \tag{43}$$

7 Experimental evaluation

In this section, we present the experiments performed to validate the presented error analysis and to compare position based and hybrid visual servoing. We begin by considering the pose estimation algorithm, then investigate the visual servoing approaches separately, and conclude by discussing the relative properties of the approaches. We have also performed experiments on the estimation of uncertainty of optimization-based pose estimation, which indicate that the estimates are valid but choose not to presents these results for the sake of brevity.

7.1 Pose estimation

The validity of the analysis was evaluated using Monte-Carlo simulations. This approach was chosen because the measurement uncertainty is a statistical phenomenon, and Monte-Carlo methods allow the assessment of such quantities. Also, the assessment of this statistical phenomenon would be very difficult using a real robot and a vision system, because it would not be possible to accurately estimate the ground truth of the robot motion to the required accuracy. Figure 3 shows the validity region of the error estimation. The deviation of the translation with respect to image error is presented on the left in Figure 3, while the deviation in the rotation angle is on the right. The image coordinates used are the pixel coordinates of a simulated camera with 512×512 pixel resolution. The lines present the predicted deviations while the crosses are the measured estimates from 1000 Monte-Carlo simulations. To generate the Monte-Carlo cases, Gaussian random noise of different variances was added to the correct point locations. The breakdown point of the error estimation is when the standard deviation in the image coordinates is approximately 5 pixels. Naturally, the breakdown point depends on the feature point configuration. The 6-feature target and its point deviations used in the experiment is shown on the left in Figure 4. Only a part of the whole 512×512 image is shown for better accessibility and the viewpoint corresponds to the initial pose of the visual servoing used later in the experiments. The longest

dimension of the target object is 0.5m and the distance is approximately 8m. Four of the feature points lie in a plane while two are displaced by a small amount. It should be noted that the breakdown point of the error estimation coincides with the breakdown point of the pose estimation, that is, the error estimation becomes invalid when the pose estimation algorithm starts to break down. An obvious restriction of the linear error estimation is its inability to predict the breakdown point as it is primarily a higher order phenomenon.

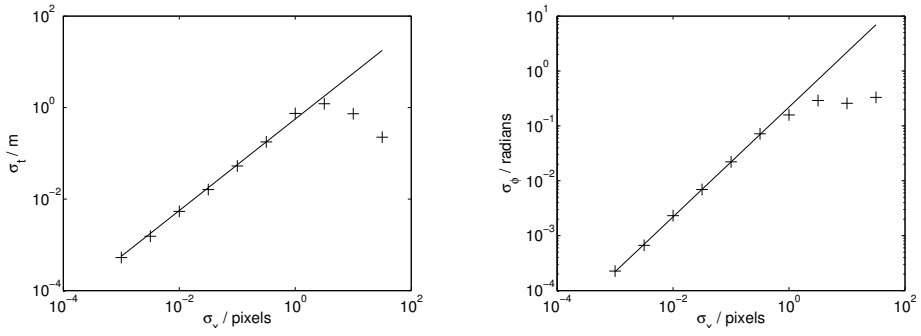

Fig. 3. Measured and predicted deviations in pose estimates with respect to image error: (left) translation; (right) rotation angle.

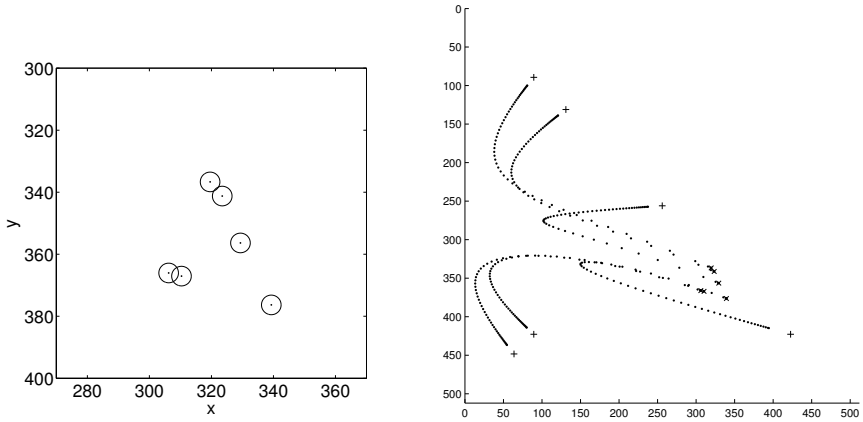

Fig. 4. (left) Deviation of image points for $\sigma_x = 10^{0.65} \approx 5$ pixels; (right) Image plane trajectories for PBVS.

The point configuration has a remarkable effect on the pose estimation accuracy. To investigate this, an experiment with random configurations was performed. Each configuration had 8 points uniformly distributed inside a cube with 1m sides. Again, we used Monte-Carlo simulations to examine the uncertainty. In Figure 5, the measured and predicted uncertainties are shown for 50 random configurations for two translational axes of freedom. The uncertainties are expressed in meters. The solid lines are the predictions and the crosses the Monte-Carlo estimates. The predictions show high correlation to the Monte-Carlo measurements, with correlation coefficients of $\rho_X \approx 0.995$ and $\rho_Y \approx 0.996$. Thus, the prediction is able to describe the effect of the point configuration. A noteworthy point is also that the peaks in the graphs in Figure 5 do not always coincide. This means that a configuration often has larger

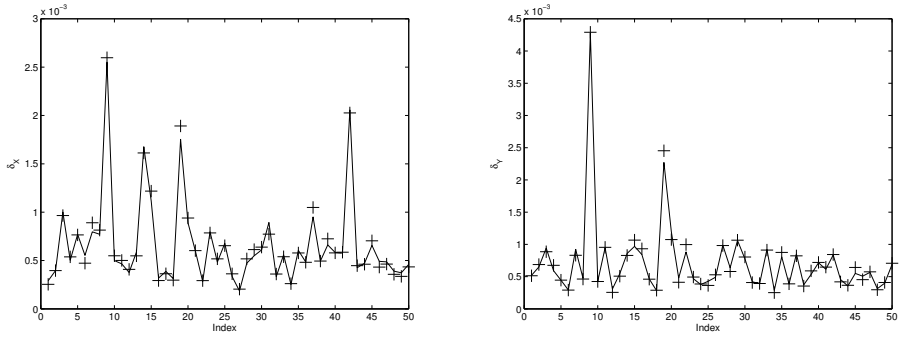

Fig. 5. Effect (uncertainty in meters) of point configuration on two axes.

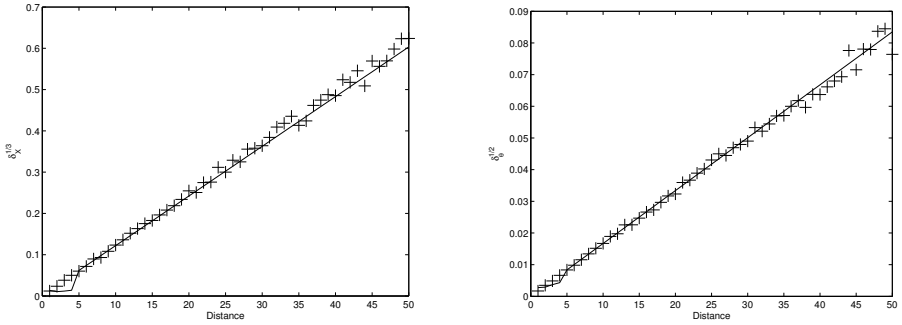

Fig. 6. Effect of distance on (left) translation; (right) rotation angle.

uncertainty in some directions than others.

Another parameter affecting the accuracy is the distance to the target. A set of Monte-Carlo experiments was performed assuming constant deviation in image coordinates. Figure 6 shows how the uncertainty grows as the distance increases. The solid lines are again the predictions and the crosses are Monte-Carlo estimates. The errors are scaled with an exponential factor to demonstrate the error behavior, with meters used for distance and radians for angle. The graph on left shows the cubic root of the translational error on one axis with respect to the distance while the right one is the square root of the error in the rotation angle. It is easy to see that the translation error is proportional to the third power of the distance while the rotational error is proportional to the squared distance. The cubic nature of the translational error was observed for the all three axes.

The analysis predicted also very high correlation between the translation in x-axis and rotation around y-axis, as well as vice versa. This translation-rotation-ambiguity is a well known phenomenon in structure-from-motion.

7.2 Position-based servoing

The following experiments assume a servoing task where the camera is initially rotated around all axes and positioned relatively far away from the goal position (around ten times the desired distance). This allows us to evaluate the effect of the distance to the servoing and also investigate the rotation around different axes. The target is the same as presented in the previous section. The trajectories of the features in the image plane are shown on the right in Figure 4.

The validity of the analysis was verified by displacing the feature locations using a known error distribution and measuring the deviation of the control output. In Figure 7, the predicted and measured deviations in the translational velocities in y (dotted line) and z (dashed line) are shown, as well as in the rotational velocities around the same axes. The units are m/s for translational velocity and $radian/s$ for rotation. The figure shows that the measured and predicted deviations correspond very well, which indicates that the theoretic analysis is valid. It should be noted that the errors are proportional to the gain of the controller λ (see Eq. 31) which was set to unit value in the experiment. Thus, the graphs should be only used to investigate the behavior over time, and the absolute values are not essential.

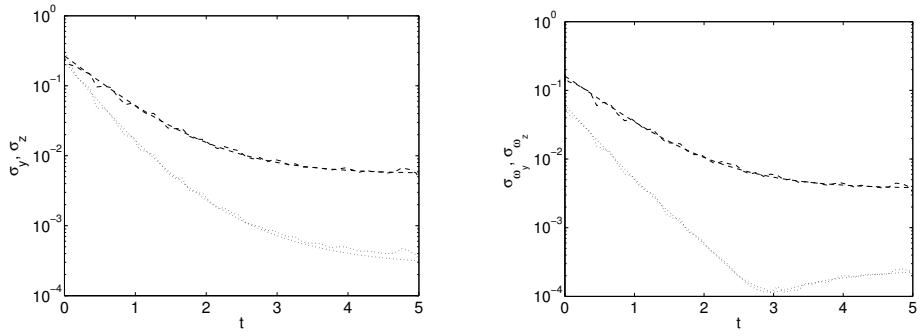

Fig. 7. Measured and predicted deviations in PBVS control output: (left) translation; (right) rotation.

Figures 8 and 9 demonstrate the error behavior of position-based servoing. The results are presented in both world and camera frames because the world frame is the most natural way to inspect the error in terms of the Cartesian controller, while the camera frame reveals information about the directional nature of the error. In the figures, the solid line corresponds to x-axis translation and rotation, the dotted line to y-axis, and the dashed line to z-axis.

Left column in Figure 8 shows the negative exponential velocity of the Cartesian control in PBVS. The absolute deviations (in m/s) in the control are presented in the middle of Figure 8, and relative (deviation divided by the control output) on right. It should be again noted that the absolute values of

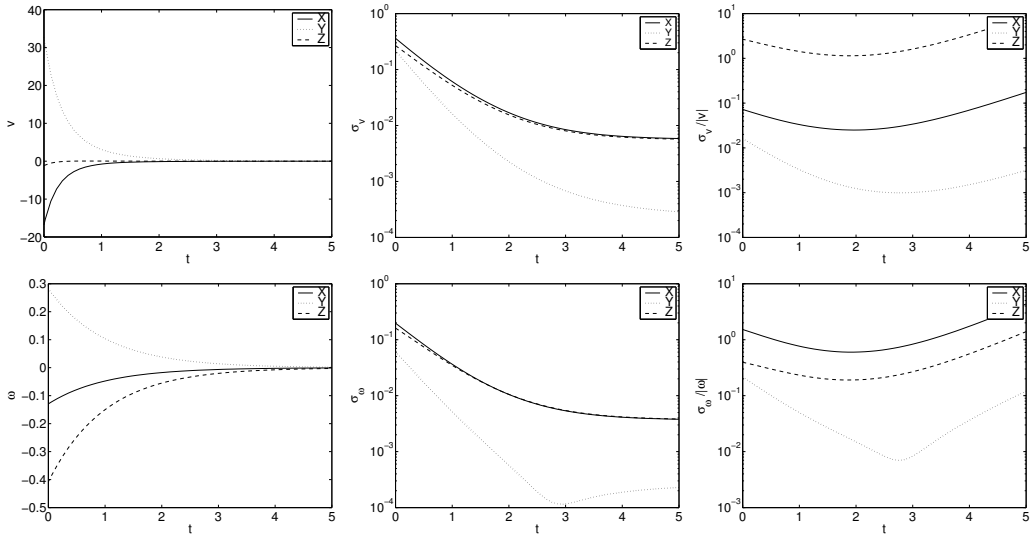

Fig. 8. PBVS behavior in world frame: (left column) Velocity/control output; (middle column) Absolute errors; (right column) Relative errors; (top row) translation; (bottom row) rotation.

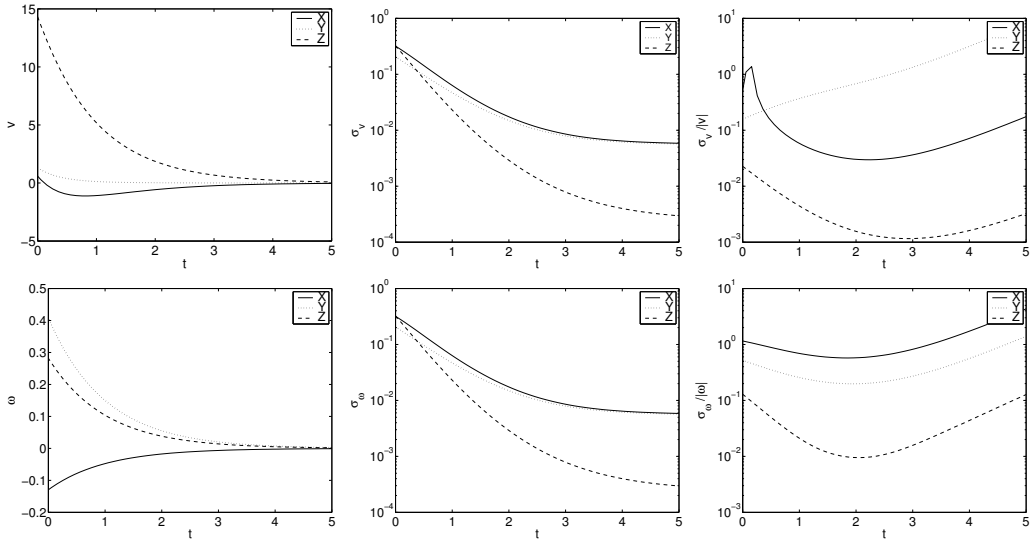

Fig. 9. PBVS behavior in camera frame: (left column) Velocity/control output; (middle column) Absolute errors; (right column) Relative errors; (top row) translation; (bottom row) rotation.

deviation are not very important as the gain of the controller affects them. In the relative errors the effect of the gain is negated as both the control output and the uncertainty are affected by the same gain. Figure 9 presents the behavior in camera frame. It can be seen that the control in the direction of the camera optical axis is much more reliable in terms of image errors, as is also the rotation around the optical axis (coinciding with the world y-axis in the goal position). There seems to be little difference in the control in the axes perpendicular to the optical axis near the goal position, but initially when the

object is not yet aligned to the image plane, there is some difference in the accuracy. The maximum shown in top right subfigure of Figure 9 is caused by the zero-crossing of the corresponding control (note that this zero-crossing occurs only in the camera frame, not in the world frame where PBVS guarantees a trajectory along a straight line). Another observation to make is that the relative error has a minimum along the trajectory, when the distance to the target is already quite small, but the target is still not precisely aligned. After this minimum, the relative error continues to increase to a level that would make the servoing impossible if the error would exist in practice.

The relative errors can be used to assess the validity of the servoing in the direction of a certain axis so that when the relative error becomes dominant (say, more than third of the control), the control in that axis can begin to diverge. The target is almost perfectly aligned in z-coordinate of the world frame, so the relative error in z-translation is very high throughout the motion, which seems to suggest that there is no reason to control that axis.

7.3 Hybrid visual servoing

The same control task used with PBVS was also used with the hybrid approach. The results of the analysis were verified by an experiment, which is presented in Figure 10. The predicted deviations seem to follow the measurements well, which suggests that the analysis is valid.

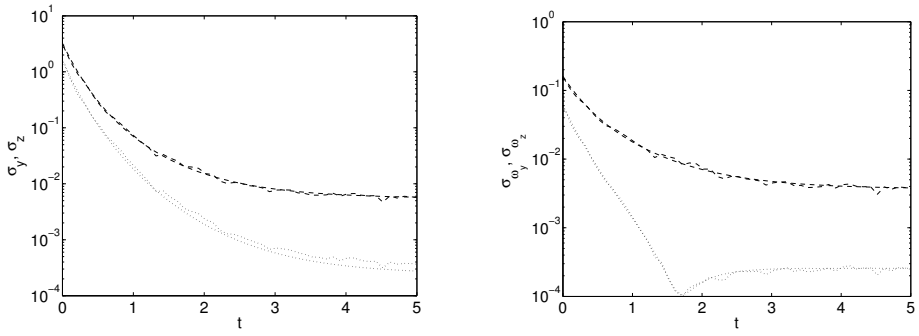

Fig. 10. Measured and predicted deviations in HYBVS control output: (left) translation; (right) rotation.

The error behavior of HYBVS can be seen in Figure 11 for the world frame and in Figure 12 for the camera frame. In some respects, the behavior is similar to PBVS. Most importantly, the errors in the translation along the optical axis and in the rotation around it are considerably smaller than for the axes parallel to the image plane. The behavior in rotation resembles that of PBVS, but it is important to note that they are not identical, as the systems have a different trajectory. It is easy to notice that HYBVS has a faster convergence in the depth (Figure. 8 and 11), and this seems to be the reason for it to

attain the constant error region of z-axis rotation sooner (Figures 8 and 11). The relative errors show another characteristic of HYBVS, the occurrence of zero-crossings in the Cartesian control. This can be seen easily from the strong peak of the relative error in translation (Figure 8). The relative errors also suggest the regions where the control is likely to diverge due to the errors in pose estimation. For HYBVS it seems that translation can be controlled at least to some degree in x and y and rotation in y and z. Now, the translation-rotation ambiguity can be again seen as the z-axis translation corresponds to x-axis rotation in the world frame.

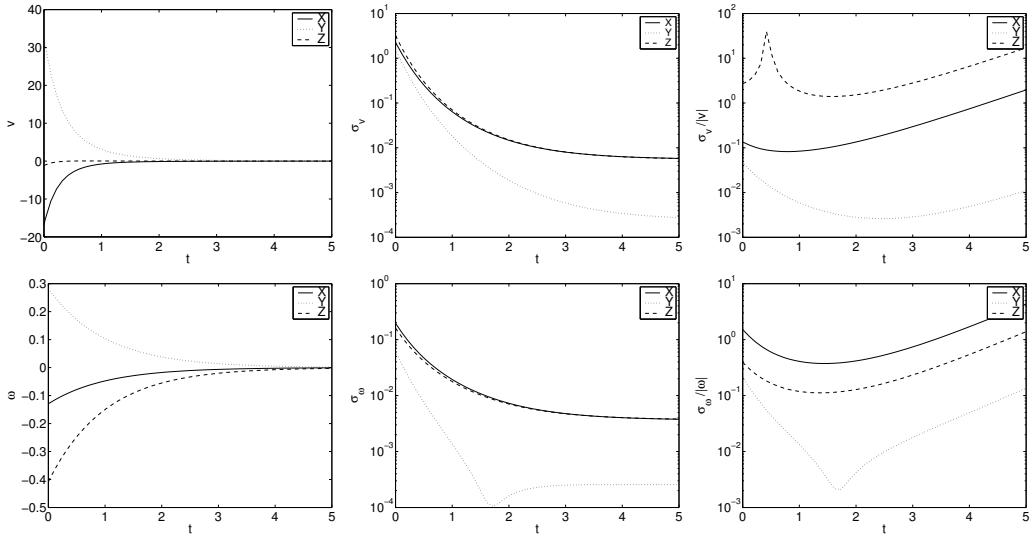

Fig. 11. HYBVS behavior in world frame: (left column) Velocity; (middle column) Absolute errors; (right column) Relative errors; (top row) translation; (bottom row) rotation.

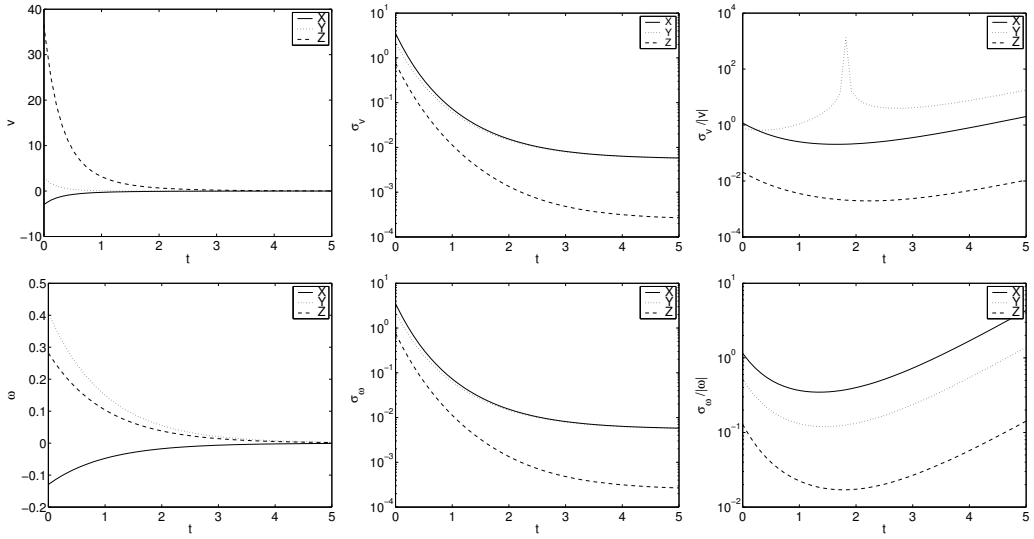

Fig. 12. HYBVS behavior in camera frame: (left column) Velocity; (middle column) Absolute errors; (right column) Relative errors; (top row) translation; (bottom row) rotation.

7.4 Discussion

A common reference trajectory needs to be defined in order to compare PBVS and HYBVS uncertainties with respect to time. Figure 13 presents the estimated errors of HYBVS control when the camera is moved along the trajectory generated using PBVS. Thus, the camera location with respect to time corresponds to Figure. 8 and 9. The errors are presented in the world frame. The absolute and relative errors for translation in Figure 13 correspond to Figure 8. For the rotation part, the errors are not shown, as they would be identical to the PBVS case since the rotation control is identical. PBVS has clearly smaller absolute errors than HYBVS in the beginning. A possible explanation for this is that the methods follow a different trajectory. Another issue is the ability of PBVS to use all feature points for the pose estimation, while HYBVS uses only a single point to control the trajectory parallel to the image plane. The relative errors have some similarities, in particular the order of the axes is the same. For the translation along y (which is closest to the optical axis, and which has the longest initial distance), the relative error amplitudes seem to be comparable. For the x-axis, which has some initial error, PBVS is initially less prone to errors, while later in the trajectory the errors become comparable. For the z-axis, HYBVS has slightly smaller error, but it is unlikely that either can be used for efficient control, as the error is large. In addition, the reason that HYBVS has smaller relative error is that it initially controls the axis away from the point of convergence as was seen in the existence of the zero-crossing discussed earlier.

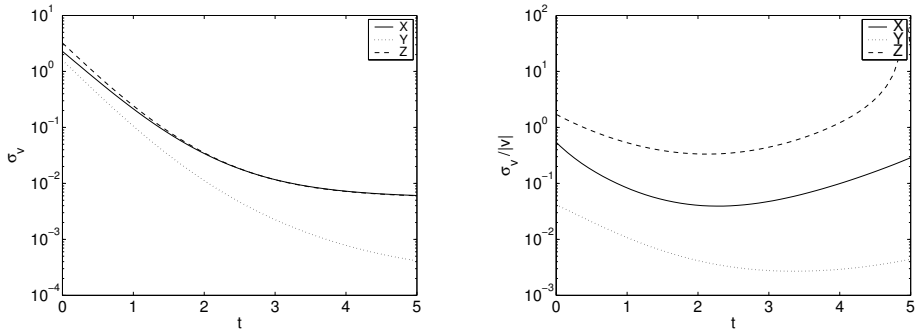

Fig. 13. HYBVS translation errors on PBVS trajectory: (left) absolute; (right) relative.

8 Summary and Conclusion

In this paper, we have analyzed the effect of measurement errors in visual servoing. The main contribution of this paper is the idea of the propagation of image error through pose estimation and visual servoing control law. In

particular, we have investigated the properties of the vision system and their effect to the performance of the control system. Two servoing approaches have been evaluated: i) position-based, and ii) 2 1/2D visual servoing. In our approach, the analysis is performed using a particular pose estimation algorithm and two visual servoing approaches. Since the general methodology can be used with any other pose estimation algorithm, we believe that this work provides novel information and serve as an example that our analysis framework is suitable for visual sevoing. It is also possible to extend the results to other servoing strategies. Particularly, we show how any optimization-based pose estimation approach can be analyzed. However, we feel that a closed-form solution is better suited to visual servoing because of real-time issues involved. We believe that our evaluation offers a valid tool to design hybrid control systems based on, for example, switching [18] or partitioning [6].

Our future work will investigate the following questions: Can we use this measure of uncertainty to control only viable degrees of freedom? For example, to first control the robot to a more reasonable distance from an initially distant pose and then, when close to target, control the more difficult degrees of freedom. Recently, a stacked controller architecture has been proposed which could be used to implement this type of control [21]. We also want to propagate the error through the pure image based visual servoing control law and compare this to the results presented here. The last question we want to answer is: Can we use this type of evaluation to find favorable feature configurations so to obtain optimal or stable behavior, especially in the case of image based visual servoing, [22].

References

[1] S. Hutchinson, G. D. Hager, P. I. Corke, A tutorial on visual servo control, IEEE Transactions on Robotics and Automation 12 (5) (1996) 651–670.

[2] D. Kragic, H. Christensen, Survey on Visual Servoing for Manipulation, Tech. Rep. ISRN KTH/NA/P–02/01–SE, January 2002, CVAP259.

[3] B. Espiau, Effect of camera calibration errors on visual servoing in robotics, in: 3rd International Symposium on Experimental Robotics, Kyoto, Japan, 1993, pp. 182–192.

[4] F. Chaumette, Potential problems of stability and convergence in image-based and position-based visual servoing, in: The Confluence of Vision and Control, no. 237 in Lecture Notes in Control and Information Sciences, Springer-Verlag, 1998, pp. 66–78.

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

[6] E. Malis, F. Chaumette, S. Boudet, 2-1/2-D visual servoing, IEEE Transactions on Robotics and Automation 15 (2) (1999) 238–250.

[7] V. Kyrki, D. Kragic, H. I. Christensen, Measurement errors in visual servoing, in: IEEE International Conference on Robotics and Automation, Vol. 2, New Orleans, LA, USA, 2004, pp. 1861–1867.

[8] M. A. Fischler, R. C. Bolles, Random sample consensus: A paradigm for model fitting with application to image analysis and automated cartography, Communications of ACM 24 (6) (1981) 381–395.

[9] L. Quan, Z. Lan, Linear n-point camera pose determination, IEEE Transactions on Pattern Analysis and Machine Intelligence 21 (8) (1999) 774–780.

[10] P. D. Fiore, Efficient linear solution of exterior orientation, IEEE Transactions on Pattern Analysis and Machine Intelligence 23 (2) (2001) 140–148.

[11] A. Ansar, K. Daniilidis, Linear pose estimation from points or lines, IEEE Transactions on Pattern Analysis and Machine Intelligence 25 (5) (2003) 578–589.

[12] B. K. P. Horn, Closed-form solution of absolute orientation using unit quaternions, Journal of the Optical Society of America 4 (4) (1987) 629–642.

[13] J. Weng, T. S. Huang, N. Ahuja, Motion and structure from two perspective views: Algorithms, error analysis, and error estimation, IEEE Transaction on Pattern Analysis and Machine Intelligence 11 (5) (1989) 451–476.

[14] R. M. Haralick, H. Joo, C.-N. Lee, X. Zhuang, V. G. Vaidya, M. B. Kim, Pose estimation from corresponding point data, IEEE Transactions on Systems, Man, and Cybernetics 19 (6) (1989) 1426–1446.

[15] L. Deng, W. J. Wilson, F. Janabi-Sharifi, Characteristics of robot visual servoing methods and target model estimation, in: Proceedings of the 2002 IEEE International Symposium on Intelligent Control, Vancouver, Canada, 2002, pp. 684–689.

[16] A. Comport, M. Pressigout, E. Marchand, F. Chaumette, A visual servoing control law that is robust to image outliers, in: Proceedings of the 2003 IEEE/RSJ Intl. Conference on Intelligent Robots and Systems, Las Vegas, Nevada, 2003, pp. 492–497.

[17] D. Kragic, H. I. Christensen, Cue integration for visual servoing, IEEE Transactions on Robotics and Automation 17 (1) (2001) 18–27.

[18] N. R. Gans, S. A. Hutchinson, An asymptotically stable switched system visual controller for eye in hand robots, in: Proceedings of the 2003 IEEE/RSJ Intl. Conference on Intelligent Robots and Systems, Las Vegas, Nevada, 2003, pp. 735–742.

[19] D. G. Lowe, Distinctive image features from scale-invariant keypoints, International Journal of Computer Vision 60 (2) (2004) 91–110.

[20] R. M. Haralick, Propagating covariance in computer vision, International Journal of Pattern Recognition and Artificial Intelligence 10 (5).

[21] N. Mansard, F. Chaumette, Tasks sequencing for visual servoing, in: IEEE/RSJ International Conference on Intelligent Robots and Systems, Vol. 1, Sendai, Japan, 2004, pp. 992–997.

[22] E. Malis, P. Rives, Robustness of image-based visual servoing with respect to depth distribution errors, in: IEEE International Conference on Robotics and Automation, 2003, pp. 1056–1061.