Robust Jacobian Matrix Estimation for Image-Based Visual Servoing

D. I. Kosmopoulos [a]

[a] *NCSR Demokritos, Inst. of Informatics and Telecommunications, GR-15310, Greece*

Abstract

A method for local estimation through training of the feature Jacobian for uncalibrated closed-loop robot manipulator control is presented, which can handle non-gaussian outliers due to illumination changes. This is achieved through the employment of a robust estimator. The method is experimentally validated through a sunroof fitting robot in a closed loop control scheme.

Key words: Visual servoing, Jacobian matrix, robust estimation, robot manipulator control, semi structured environments.

1 Introduction

The integration of visual sensors in robotic systems can be very beneficial for large scale industrial production, due to the high precision that can be achieved with vision (e.g., [1], [2]). A popular scheme for visual servo-controlled robotic systems is the dynamic look and move; the related systems can be divided to position-based or image-based [3] [4].

In the position-based systems the manipulator is regulated after calculation of the end-effector pose in the workspace coordinates; the extracted features from the image are used for fitting a geometric model of the target (and sometimes of the end-effector) to estimate the relative pose of the target with respect to the end-effector. This method requires camera calibration with respect to the robot.

The image-based systems use directly the image features space to find the desired end-effector pose by comparing the current visual features with the

Email address: dkosmo@iit.demokritos.gr (D. I. Kosmopoulos).

visual features observed when the end-effector has its desired pose. The latter method although less intuitive, is in many cases preferable. It eliminates some drawbacks of the position-based approach and namely the need for the computationally complex depth estimation and for the time-consuming, and very often inaccurate, sensor calibration.

Although it is known that the position-based methods perform generally better when the displacement from the goal is large, the image-based techniques are more precise for small errors [5], [6] and therefore, the image-based techniques are more appropriate for structured production lines with relatively small perturbations. It is possible to combine the advantages of each method by dynamically switching between them, based on proximity to target [6]. Other approaches adopt hybrid state vectors composed by both 2-D and 3-D visual measurements, referred to as 2.5-D visual servoing [7]. The reader can find a detailed discussion on hybrid methods in [8]. Such methods are obviously expected to benefit from improvements in the two basic approaches.

As for the image-based servoing systems, their main characteristic is that they directly associate the pose error of the end-effector with an image Jacobian matrix; this matrix models the variation of selected image features (on the target or the end-effector) with respect to the difference of the current end-effector pose from its nominal pose. By solving the inverse problem it is possible from the differences in the image feature vector to calculate the desired end-effector pose. An overview of the control scheme is presented in Figure 1.

Several works following the image-based approach have been presented in the past such as [9] and [10]. However, in those and other similar works the presented Jacobian matrices actually depend on the depth information, thus minimizing the benefits of the image-based approach, as has been pointed out in [11]. Although this dependence is not a problem for planar workspaces with constant depth, (e.g., [9]), it introduces additional overhead in workspaces where the depth may vary. The reader is referred to [5] for a detailed discussion on this issue.

Further to the above works in [12] uncalibrated visual servoing for static targets using fixed cameras was presented. In [13] has improved the control scheme to eye-in-hand stereo tracking of moving targets using static reference points to estimate the target motion, through the real-time estimation of Jacobian matrix. In [14] the Jacobian matrix estimation was expanded with convergence analysis. In [15] the positioning of a 6-DOF robot camera with respect to a target object is handled assuming affine perspective transformation. However, in these works perfect feature extraction is assumed, which is not the case in many realistic environments, where outliers can cause control failures. The instability due to outliers can be a major conern when installing cameras in industrial environments.

Recently efforts have been made towards approaches that do not require a-priori model of the target such as in [16], [17]. However, in a structured industrial environment the types of objects to be manipulated are usually known in advance and the overhead of learning offline their models is considered worthwhile, especially when high accuracy is required.

In this work we present a method for local calculation of the image Jacobian through training without the need for depth estimation, thus exploiting to the full extend the advantages of image-based visual servoing. The proposed method is robust to outliers, thus handles the inevitable imperfections from image processing mainly due to illumination variations, e.g., welding sparks. The approach can be easily applied in semi-structured environments, such as in production lines, where relatively small, nevertheless critical, deviations from the end-effector nominal pose may occur.

The rest of the paper is organized as follows: the next section describes the context of using the image Jacobian for visual servoing tasks; section 3 provides the details for the calculation of the image Jacobian in the subspace where the robot executes the visually controlled task; section 4 provides experimental results for a sunroof fitting prototype robotic installation that utilizes visual feedback and the paper concludes with section 6.

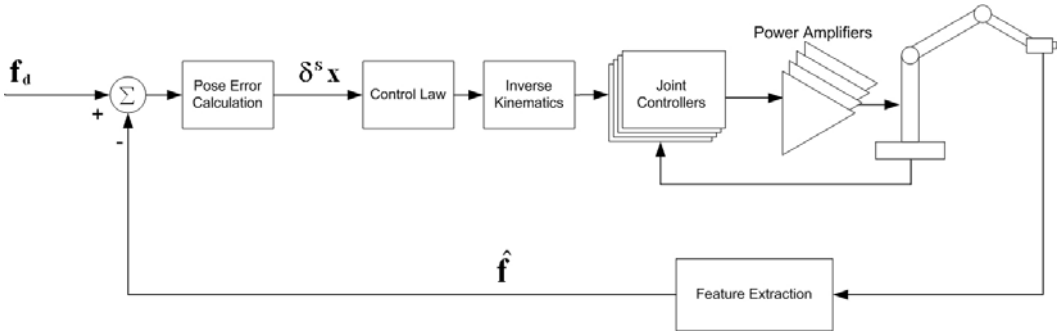


Fig. 1. Image based dynamic "look and move" system. $\delta^s\mathbf{x}$ is the end-effector-pose correction vector, which is the result of subtracting the measured feature vector $\widehat{\mathbf{f}}$ from the desired features vector $\mathbf{f}_d$.

2 Uncalibrated image-based visual servoing

The control task in the case of uncalibrated image-based visual servoing is described in Figure 1. The features extracted from the camera image are the variables of an error function, which depends on the relative pose of the depicted objects to the camera. Making this error function zero is actually equivalent to achieving the desired task. Generally this function is non-linear and cross-coupled, which means that the motion along one degree of freedom of the robot will result in a complex motion of many features in the image. However,

in a small workspace this function can be approximated in a linear fashion through the use of the image features Jacobian matrix. Similarly to ([3]) this can be formalized in the following.

A task in the m-dimensional feature space can be described by an error function:

$$\mathbf{E}(\mathbf{C}_e({}^w\mathbf{x}_e), \mathbf{C}_t({}^w\mathbf{x}_t)) : C_e \times C_t \rightarrow \Re^d \tag{1}$$

where C_e, C_t, the end-effector and target workspaces correspondingly ($\mathbf{C}_e \in C_e$, $\mathbf{C}_t \in C_t$, $d \leq m$), d the workspace dimensionality and ${}^w\mathbf{x}_e$, ${}^w\mathbf{x}_t$ the end-effector (robot tool) and target (object to be handled) pose vectors in the world coordinate system (in general by the term ${}^a\mathbf{x}_b$ we will denote the relative pose vector of the coordinate system attached to body b with respect to the coordinate system attached to body a). The end-effector has the desired pose when $\mathbf{E}({}^w\mathbf{x}_e, {}^w\mathbf{x}_t)$=0.

Let us define an image error function:

$$\mathbf{e} : F_e \times F_t \rightarrow \Re^d \tag{2}$$

where F_e, F_t the image features spaces belonging to the end-effector and to the target which result from projection of C_e, C_t to the camera plane. The $\mathbf{e}$ is equivalent to an error function in the d-dimensional workspace ($d \leq m$) if:

$$\mathbf{E}(C_e({}^w\mathbf{x}_e), C_t({}^w\mathbf{x}_t)) = 0 \Leftrightarrow \mathbf{e}(\mathbf{f}_e({}^c\mathbf{x}_e), \mathbf{f}_t({}^c\mathbf{x}_t)) = 0 \tag{3}$$

for all $\mathbf{f}_e \in F_e$, $\mathbf{f}_t \in F_t$, except for some singularity points (${}^c\mathbf{x}_e$,${}^c\mathbf{x}_t$ the end-effector and the target pose vectors in the camera coordinate system correspondingly).

The association of $\mathbf{e}$ with differential changes of the end-effector pose is done assuming linearity through the Jacobian matrix $\mathbf{J}$ given by the equation:

$$\mathbf{e}(\mathbf{f}_e, \mathbf{f}_t) = \mathbf{J}({}^e\mathbf{x}_t) \cdot \delta^e\mathbf{x}_t \tag{4}$$

where $\delta^e\mathbf{x}_t$ is the end-effector pose error and $\mathbf{J}$ is defined as:

$$\mathbf{J}({}^e\mathbf{x}_t) = \frac{\partial \mathbf{e}(\mathbf{f}_e, \mathbf{f}_t)}{\partial^e\mathbf{x}_t} \tag{5}$$

where ${}^e\mathbf{x}_t$ is the pose vector of the target in the end-effector coordinate system.

The current pose error $\delta^e\mathbf{x}_t$ is used for correcting the pose of the end-effector. Given $\mathbf{e}(\mathbf{f}_e, \mathbf{f}_t)$, it is calculated by:

$$\delta^e\mathbf{x}_t = \mathbf{J}^+\mathbf{e}(\mathbf{f}_e, \mathbf{f}_t) \tag{6}$$

In the above equation $\mathbf{J}^+$ is the generalized inverse of $\mathbf{J}$ and is given by:

$$\mathbf{J}^+ = \begin{cases} \mathbf{J}^T \cdot (\mathbf{J} \cdot \mathbf{J}^T), d > m \\ (\mathbf{J}^T \cdot \mathbf{J})^{-1} \cdot \mathbf{J}^T, d < m \\ \mathbf{J}^{-1}, d = m \end{cases} \tag{7}$$

As an example, in the case that the cameras are mounted on the manipulator and only the target is monitored ($\mathbf{f}_e = \mathbf{0}$) the $\mathbf{J}$ is given by [9]) :

$$\mathbf{J} = \begin{bmatrix} \frac{\alpha_x f}{f - {}^c z_i} & 0 & \frac{\alpha_x f {}^c x_i}{(f - {}^c z_i)^2} & \frac{\alpha_x f {}^c x_i {}^c y_i}{(f - {}^c z_i)^2} & \frac{\alpha_x f}{f - {}^c z_i}({}^c z_i - \frac{{}^c x_i^2}{f - {}^c z_i}) & -\frac{\alpha_x f {}^c y_i}{f - {}^c z_i} \\ 0 & \frac{\alpha_y f}{f - {}^c z_i} & \frac{\alpha_y f {}^c y_i}{(f - {}^c z_i)^2} & \frac{\alpha_y f}{f - {}^c z_i}(-{}^c z_i + \frac{{}^c y_i^2}{f - {}^c z_i}) & \frac{-\alpha_y f {}^c x_i {}^c y_i}{(f - {}^c z_i)^2} & -\frac{\alpha_y f {}^c x_i}{f - {}^c z_i} \end{bmatrix} \tag{8}$$

where ${}^c x_i$, ${}^c y_i$, ${}^c z_i$ are the coordinates of a target point in the camera coordinate system, α_x, α_y the image scaling factors in x and y direction and f the camera focal length. Obviously there is dependence on the relative position of the camera to the target as well as on the camera parameters α_x, α_y. This requires calculation of 3D position and camera calibration, which partially cancels the advantages of the image-based approach; alternatively a CAD model of the target can be used, which however, when available, could lead to rather complex optimization problems [9].

In the following we propose an outlier-tolerant training method for building a model of the end-effector pose with respect to image-features in order to execute a visual servoing task. Assuming operation in a small workspace volume we do not need to estimate the depth, and we do not need sensor calibration. Thus the method maintains all the advantages of a pure image-based approach.

3 Jacobian matrix estimation

For many industrial applications the workcell is rather well-structured with relatively small tolerances and the manipulated target may appear within a relatively small volume. The center of this volume is usually as the nominal

target pose, where the positioning error is zero; the size of this volume is defined by the target tolerances in position and orientation.

For small volumes of interest it is reasonable to assume that the Jacobian matrix remains constant and therefore it can be calculated through an offline training procedure. The task execution after the training does not require a-priori knowledge of the relative pose between sensor and target. The robotic system can be trained for execution of several robotic tasks thus acquiring the knowledge of how to execute them. This knowledge can be stored and retrieved before the execution of the respective task, to exploit sensory feedback information. This kind of training is normaly repeated on a regular basis with target objects of accurately known pose and dimensions, to alleviate changes in their dynamic characteristics or in precision due to extensive industrial use.

The training starts with the end-effector going to the desired pose $\mathbf{x}_d$ that is the nominal pose relative to the target, e.g., in the case of a hole drill task the end-effector is positioned over the desired hole position. Then the sensors are activated and through feature extraction the nominal feature vector $\mathbf{f}_d$ is calculated. We assume that in a previous stage we have selected appropriate features which are not ambiguous, can be located easily under different views and represent projections of physical structures to the image [3], [4]. However, even carefully selected features may still have problems to calculate due to noise, feature extraction imperfections or illumination changes. Clearly there is a need for a framework able to minimize the influence of outliers in visual measurements.

Assuming a manipulator with six degrees of freedom, i.e., x, y and z for position and a, b, c for orientation, and m features extracted from image data then the Jacobian matrix will be given by:

$$\mathbf{J} = \begin{bmatrix} \frac{\partial f_1}{\partial x} & \frac{\partial f_1}{\partial y} & \frac{\partial f_1}{\partial z} & \frac{\partial f_1}{\partial a} & \frac{\partial f_1}{\partial b} & \frac{\partial f_1}{\partial c} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \frac{\partial f_m}{\partial x} & \frac{\partial f_m}{\partial y} & \frac{\partial f_m}{\partial z} & \frac{\partial f_m}{\partial a} & \frac{\partial f_m}{\partial b} & \frac{\partial f_m}{\partial c} \end{bmatrix} \tag{9}$$

For calculating the $\mathbf{J}$, the end-effector is positioned in k predefined consecutive spatial poses (normally in equal intervals) around the desired (nominal) pose $\mathbf{x}_d$ changing only the r-th degree of freedom, where $r \in \{x, y, z, a, b, c\}$ while the position and orientation with respect to the other degrees of freedom does not change. In each intermediate position r_i of the current degree of freedom, input from the sensors is acquired and the feature vector $\mathbf{f}_s(r_i)$ is measured (for features s=1, 2,. . . ,m and for intermediate positions r_i for i=1,2,. . . , k). In the relatively tight volume that we consider here, the coefficients of the corresponding column of $\mathbf{J}$ can be assumed constant and can be calculated by

exploiting the feature vector measurements $\mathbf{f}_s(r_i)$. In each training pose r_i all the coefficients of the respective column are calculated simultaneously. The same procedure is repeated for all columns.

Let us examine each $\mathbf{J}$ coefficient $\partial f_s/\partial r$ separately. To cope with the outliers that are due to noise in feature measurements (and can corrupt the estimated $\mathbf{J}$, leading the system to instability) we employ robust statistics, which "is proper for recovering the structure that best fits the majority of the data while identifying and rejecting outliers or deviating substructures" [18].

The estimation of $\mathbf{J}$ coefficients is formulated as fitting the vector $\mathbf{A} = [\alpha_1, \alpha_2]^T$, for a linear feature model

$$\phi_{\mathrm{s}}(\mathbf{A}, \mathrm{r_i}) = \alpha_1 \mathrm{r_i} + \alpha_2 \tag{10}$$

to a set $\mathbf{M}$ of k training feature measurements for the individual feature f_s:

$$\mathbf{M} = \{f_s(r_1), f_s(r_2), \ldots, f_s(r_k)\} \tag{11}$$

The respective coefficient of $\mathbf{J}$ is given by the α_1 parameter of the feature model ϕ_s. The estimation of $\mathbf{A}$ is generally performed by minimizing the quantity:

$$\mathrm{E} = \sum_{i=1..k} \rho(g_i, \sigma) \tag{12}$$

where ρ is an estimation function and σ is a scale parameter of this function. Furthermore, g_i is the residual error value of the measurement with respect to the coefficient model, in other words:

$$g_i \equiv g(\mathbf{A}, r_i) = f_s(r_i) - \phi_s(\mathbf{A}, r_i) \tag{13}$$

If the measurement errors are normally distributed the optimal ρ function is the quadratic, which gives rise to the standard least squares estimation:

$$\rho(g_i, \sigma) = \frac{g_i^2}{2\sigma^2} \tag{14}$$

The problem with the least squares method is that too high weights are assigned to the outliers. One way to see this is by considering an influence function ψ, which characterizes the bias to the solution by a particular measurement. The derivative of the ρ function can be such a function ψ. Clearly, in the least squares case (Figure 3a) the influence of data points increases linearly without bound (Figure 3b). Obviously, to increase robustness the ψ function must minimize the influence of outliers. Such a *psi* function is the

derivative of the Geman McClure robust estimator, which has the additional advantage of gradual transition between inliers and outliers. The related ρ and ψ functions are displayed in Figures 3c-d and are given by [19]:

$$\rho(x,\sigma) = \frac{x^2}{x^2+\sigma} \tag{15}$$

$$\psi(x,\sigma) = \frac{2x\sigma}{(x^2+\sigma)^2} \tag{16}$$

Small values for scale parameter σ make the system more tolerant to outliers, since their effect is attenuated. This attenuation is depicted in Figure 3.

It can be proved ([20]) that under concavity of $q(x) \equiv \rho(\sqrt{x})$, which is here the case, any multidimensional minimization problem of the form:

$$\arg\min_{\mathbf{A}} \sum_{i=1..k} \rho(g(\mathbf{A}, r_i))) \tag{17}$$

can be turned into a dual minimization problem:

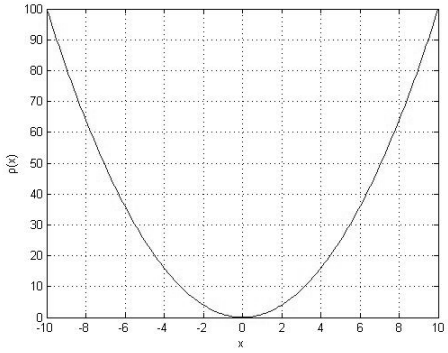
$$\arg\min_{\mathbf{A},z_i} \sum_{i=1..k} [mz_i g(\mathbf{A}, r_i)^2 + \psi(z_i)] \tag{18}$$

involving weights z_i, which lie in (0,1]. If a residual g_i is high (an outlier) the respective weight z_i should be close to zero, while for a small residual g_i (inlier, supported by the model) the z_i should be close to one. ψ is a continuous differentiable function and $m = \lim_{x->0^+} q'(x)$. In our case $m = \sigma^{-1}$.

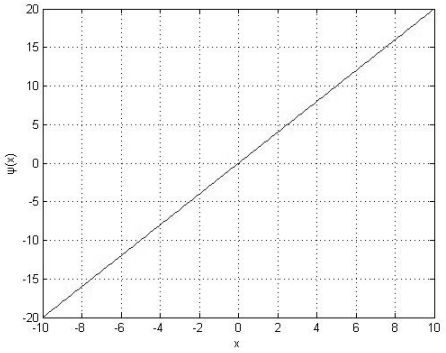
Because both the weights z_i and the residuals g_i are unknown, there is no analytical solution and the minimization has to be solved iteratively. Initially the value for all z_i is set to 1. By doing so, the dual minimization problem becomes a typical least square problem with respect to $\mathbf{A}$ (ψ gives zero in differentiation). In the next step the current value for $\mathbf{A}$ is considered constant, and thus the g_i are constant as well; then we calculate the optimal values for z_i, which are given in a simple closed form by the equation:

$$z_i = \frac{\rho'(g_i)}{2mg_i} = \frac{2g_i\sigma}{(g_i^2+\sigma)^2} \cdot \frac{1}{2\sigma^{-1}g_i} = \frac{\sigma^2}{(g_i^2+\sigma)^2} \tag{19}$$

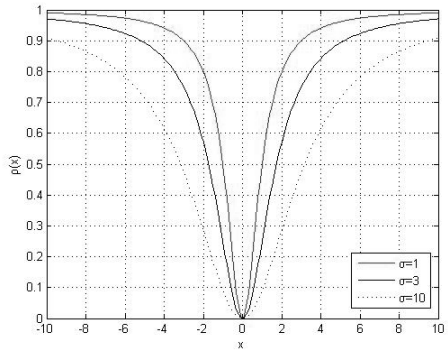
The same calculations are repeated in this order until the consecutively calculated values for $\mathbf{A}$ and z_i converge. The reader is referred to [20], [21] for more details on how (19) results from (18).



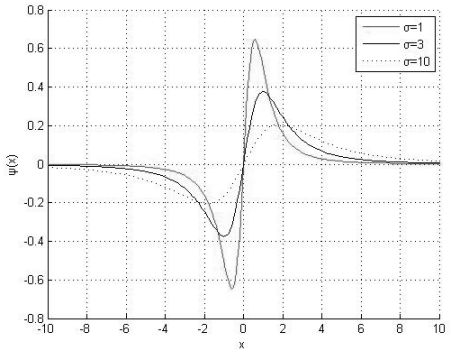
(a) $\rho(x) = x^2$



(b) $\psi(x) = 2x$



(c) $\rho(x) = \frac{x^2}{x^2+\sigma}$



(d) $\psi(x) = \frac{2x\sigma}{(x^2+\sigma)^2}$

Fig. 2. The attenuation effect of the Geman McClure ρ function compared to the quadratic ρ function. (a) and (b) display the ρ and ψ for the quadratic case, while (c) and (d) illustrate the corresponding fucntions in the Geman McClure case. The attenuation is proportional to the respective ψ function. Unlike in (d), in (b) the influence of outliers is unbounded. Also, the lower the σ the more attenuated become the outliers (responses for x's "far" from zero), as displayed in (d).

The Algorithm 1 gives an overview the proposed approach. The output is a model for each coefficient of **J** with minimum influence from outliers, since we assign low weights to them.

4 Experimental results

The applicability of the method has been verified through robust estimation of the image Jacobian for an industrial visual servoing application. It concerns a sunroof placement installation, which aims to fit a sunroof onto a car body on

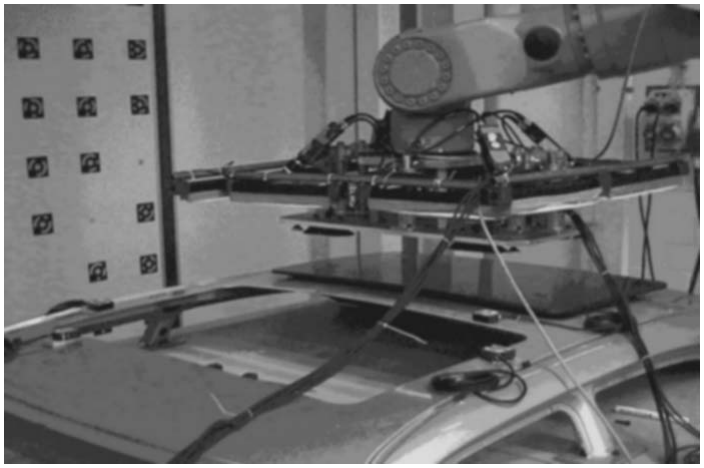


Fig. 3. Experimental setup for sunroof fitting application by employing four cameras.

Algorithm 1 Calculate the **J** coefficients for one column

```
{for all poses along a DOF}
for i = 1 to k do
  {for all feature measurements}
  for s = 1 to m do
    acquire measurement f_s(r_i) from training pose r_i
  end for
end for
for all coefficients in current column of J do
  for i = 0 to k do
    z_i=1
  end for
  repeat
    set z_i value to (18)
    solve (18) for A using least squares
    set A value to (19)
    calculate z_i using (19)
  until z_i and A converge
end for
```

the automobile production line (Figure 3). We used the 6-DOF manipulator K-125 of KUKA with the KRC-1 controller, which permits the employment of external software for control at the end-effector level.

The task of fitting the sunroof on the car body was performed using four CCD cameras, monitoring the four corners of the sunroof opening. The corners were identified as the intersection points of the monitored edges. The features used were the x, y image coordinates of the four sunroof opening corners. We have used eight image features in total for six degrees of freedom, which made an overdetermined system. Thus $\mathbf{f}_d = (x_1, y_1, x_2, y_2, x_3, y_3, x_4, y_4)$.

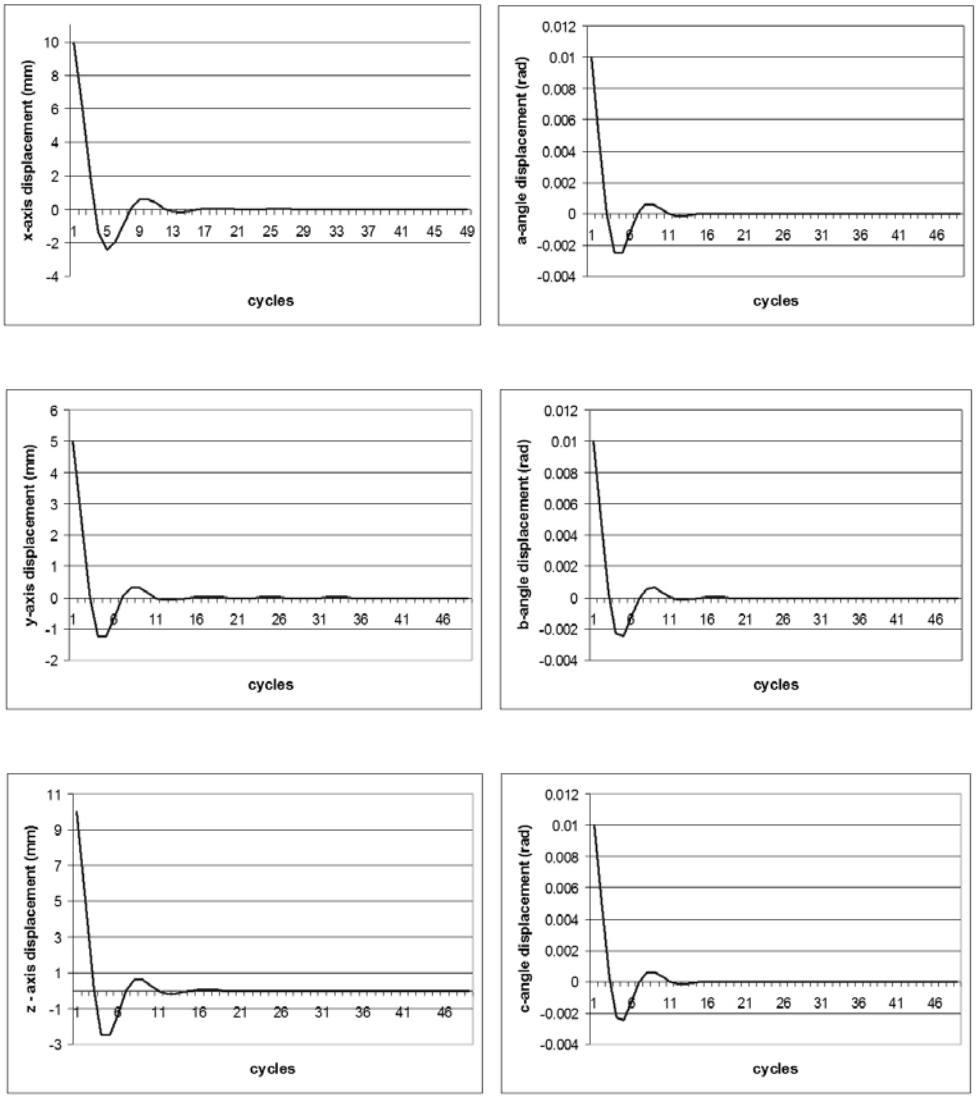


Fig. 4. System convergence for all degrees of freedom after estimation of the feature Jacobian matrix with initial pose error $(x, y, z, a, b, c) = (10, 5, 10, 0.01, 0.01, 0.01)$. The displacements for x, y, z are in mm and for a, b, c in rad. The time is measured in robot cycles (12ms).

We have trained all six degrees of freedom x, y, z, a, b, c of the manipulator. The distances for which we have trained the system (production tolerances) were 20mm for x, y, z with steps of 1mm and 5 degrees for a, b, c at steps of 0.5 degrees. We have used a lamp to distract the corner tracker and create several outliers. This is a situation that can be met in semi-structured environments, where the illumination can change, e.g., due to welding sparks. The robust estimator proved itself very useful in all cases that noisy edges were falsely identified as the target image corners. Similarly, its use can be vital in cases of not robust feature extractors or changing illumination, where outliers are expected.

Some typical system simultaneous responses for all degrees of freedom are displayed in Figure 4 using a typical PID controler. The end-effector was displaced from the nominal pose according to the vector $(x, y, z, a, b, c) =$ (10mm, 5mm, 10mm, 0.01rad, 0.01rad, 0.01rad). The Jacobian matrix that has been calculated with robust methods led the system to convergence in less than 20 robot cycles (240ms) for all degrees of freedom at the presence of noisy measurements. The steady-state error was approximately 0.1mm and 0.0002rad for distance and angle correspondingly. In cases where the robust estimator was not used, the convergence to the target could not be guaranteed for the affected degrees of freedom, because of the wrong calculation of the coefficients of the image Jacobian.

To further verify the robustness the support for the fitted model was used. In other words the extracted model was rejected if not enough samples could explain it. This was verified by the sum of the weights of the support samples which had to be higher than a threshold. Otherwise no training was possible.

The performance of the method depends on how successful is the elimination of the effect from the outliers. The significance of the parameter σ of the robust estimator is therefore high. In our epxeriments it was empirically set and was adapted to the expected standard deviation of the feature extractor under normal conditions, so that the samples that would fall out of that would have very small weight.

5 Conclusion

We have presented a method for robust estimation of the feature Jacobian matrix for visual servo controlled systems which does not require calibration. The method allows for independence of depth measurements and can be applied for task - execution in semi - structured environments, where the tolerances in target pose lie within a relatively small volume. The employment of robust statistics provides a reliable estimation of the image Jacobian even at the presence of outliers, which are due to image feature extraction imperfection, or because of illumination changes during the training procedure.

References

[1] H. Golnabi, A. Asadpour, Design and application of industrial machine vision systems, Robotics and Computer-Integrated Manufacturing 23 (6) (2007) 630 – 637.

[2] G. M. Bone, D. Capson, Vision-guided fixtureless assembly of automotive components, Robotics and Computer-Integrated Manufacturing 19 (1-2) (2003) 79 – 87.

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

[4] F. Janabi-Sharifi, W. Wilson, Automatic selection of image features for visual servoing, IEEE Transactions on Robotics and Automation 13 (6) (1997) 890–903.

[5] F. Chaumette, S. Hutchinson, Visual servo control, Part I: Basic approaches, IEEE Robotics and Automation Magazine 13 (4) (2006) 82–90.

[6] M. Staniak, C. Zielinski, Structures of visual servos, Robotics and Autonomous Systems In Press, Corrected Proof (2010) –.

[7] E. Malis, F. Chaumette, S. Boudet, 2.5-D visual servoing, IEEE Transactions on Robotics and Automation 15 (2) (1999) 238 –250.

[8] F. Chaumette, S. Hutchinson, Visual servo control, Part II: Advanced approaches, IEEE Robotics and Automation Magazine 14 (1) (2007) 109–118.

[9] J. T. Feddema, C. S. George Lee, O. R. Mitchell, Model-based visual feedback control for a hand-eye coordinated robotic system, Computer 25 (8) (1992) 21–31.

[10] Y. Mezouar, F. Chaumette, Optimal camera trajectory with image-based control, International Journal of Robotics Research 22 (10-11) (2003) 781–804.

[11] R. Horaud, F. Dornaika, B. Espiau, Visually guided object grasping, IEEE Transactions on Robotics and Automation 14 (1995) 525–532.

[12] K. Hosoda, K. Igarashi, M. Asada, Adaptive hybrid control for visual and force servoing in an unknown environment, IEEE Robotics and Automation Magazine 5 (4) (1998) 39–43.

[13] M. Asada, T. Tanaka, K. Hosoda, Visual tracking of unknown moving object by adaptive binocular visual servoing, in: Multisensor Fusion and Integration for Intelligent Systems, 1999. MFI '99. Proceedings. 1999 IEEE/SICE/RSJ International Conference on, 1999, pp. 249–254.

[14] J. Piepmeier, G. McMurray, H. Lipkin, Uncalibrated dynamic visual servoing, IEEE Transactions on Robotics and Automation 20 (1) (2004) 143–147.

[15] F. Conticelli, B. Allotta, Robust stabilization of second-order image-based affine systems, Systems and Control Letters 39 (4) (2000) 245 – 253.

[16] R. Mebarki, A. Krupa, F. Chaumette, 2-D ultrasound probe complete guidance by visual servoing using image moments, IEEE Transactions on Robotics 26 (2) (2010) 296–306.

[17] C. Collewet, F. Chaumette, Visual servoing based on structure from controlled motion or on robust statistics, IEEE Transactions on Robotics 24 (2) (2008) 318–330.

[18] F. Hampel, E. Ronchetti, R. P.J., W. Stahel, Robust Statistics: The Approach Based on Influence Functions, J.Wiley, New York, 2005.

[19] S. Geman, D. E. McClure, Statistical methods for tomographic image reconstruction, Bulletin of the International Statistical Institute 52 (1987) 5–21.

[20] E. Memin, P. Perez, Dense estimation and object-based segmentation of the optical flow with robust techniques, IEEE Transactions on Image Processing 7 (5) (1998) 703–719.

[21] P. Charbonnier, L. Blanc-Feraud, G. Aubert, M. Barlaud, Deterministic edge-preserving regularization in computed imaging, IEEE Transactions on Image Processing 6 (1997) 298–311.