Proceedings of the 1997 IEEE
International Conference on Robotics and Automation
Albuquerque, New Mexico - April 1997

Real-Time Visual Tracking of 3-D Objects with Dynamic Handling of Occlusion

P. Wunsch and G. Hirzinger

German Aerospace Research Establishment - DLR
Institute for Robotics and System Dynamics
82230 Wessling, Germany
E-mail: Patrick.Wunsch@dlr.de

Abstract — *Position-based visual servoing requires estimating and tracking the three dimensional position and orientation of a 3-D target object from camera images. This paper describes a novel approach to the problem that consists of two steps: First, a set of spatial pose constraints is derived from image features, by means of which 3-D object pose is calculated with an efficient model-fitting algorithm. Kalman-filtering is then used to estimate object velocity and acceleration. Compared to previous approaches that use Kalman-filters to directly estimate the object state from image features, the proposed method has a variety of advantages: Computation time is only $\mathcal{O}(n)$ rather than $\mathcal{O}(n^3)$ where n is the number of image features considered, sensor fusion is simplified and temporal estimation is decoupled from the choice of image features. The last point is of particular importance if occlusions that may occur during tracking are to be predicted and dynamically handled. With the tracking method proposed, a robot could be precisely controlled with respect to static objects and robustly follow targets moving in 6 degrees of freedom, while occlusions were continuously predicted and appropriate features automatically selected at video rate (25Hz). High robustness is obtained by Hough transform-based feature extraction.*

1 Introduction

The closed-loop position control of a robot end-effector based on feedback of visual measurements is commonly refered to as *visual servoing*. According to Sanderson and Weiss [13] two major categories of systems can be distinguished. In *image-based* servoing control feedback is computed by directly reducing the error between a set of image features and their predefined desired positions in the image plane. The *position-based* approach on the other hand uses a geometric model of the target in conjunction with a known model of the camera to estimate the three dimensional position and orientation of the observed object. Feedback is then computed such that an error in 3-D pose space is reduced. See [8] for a comprehensive review of state-of-the-art methods and an extensive bibliography.

Although image-based approaches usually require little computation and guarantee robustness against errors in sensor modelling and camera calibration, position-based methods have the distinct advantage that both *geometric* and *dynamic models* can be included in a straightforward fashion to increase the robustness of the vision task. Specifically, dynamic models are especially helpful in compensating computational delay and in stabelizing the estimation of velocities and accelerations, while geometric information can be used to predict occlusions that may occur during tracking and to dynamically select features that ensure well-conditioned pose estimation. However, standard recursive pose estimation methods that are usually used for position-based visual servoing require significant computation, which is further increased if features are to be dymanically reselected as will be shown in the next section. Hence, especially if measurements from multiple sensors are to be considered (sensor fusion), position-based approaches that dynamically account for occlusion may fail to achieve servo rates high enough to ensure stable control. Therefore, we propose in the following a highly efficient model-based method for estimating the three-dimensional position and orientation of a target object from a sequence of images. Apart from processing speed, the important point is that at each time step features may be selected *independently* of those chosen in previous images, which is essential for dynamic occlusion handling.

2 Previous Research

Most methods for 3-D pose estimation from an image sequence that have been used for visual servoing tasks [5, 4, 6, 14] are based on the dynamic vision approach introduced by Dickmanns [3], which is based on Kalman-filtering. The state of the target

$$\boldsymbol{x}_i = (\boldsymbol{\theta}_i, \dot{\boldsymbol{\theta}}_i, \ddot{\boldsymbol{\theta}}_i, \cdots)'$$

is represented by its 3-D pose $\boldsymbol{\theta}$, and – depending on the model of motion – one or more of its derivatives. The vector $\boldsymbol{\theta} = (x, y, z, \phi, \psi, \tau)'$ consists of three parameters describing translation of the target and another three for orientation. The relation between an image feature vector $\boldsymbol{f}_i$ and the object state $\boldsymbol{x}_i$ in the ith time step is modelled by the linear *measurement equation*

$$\boldsymbol{f}_i = \boldsymbol{H}_i \boldsymbol{x}_i + \boldsymbol{\eta}_i. \tag{1}$$

Here matrix $\boldsymbol{H}_i$ is a linearization of the perspective camera mapping, possibly including lens distortions, and hence a function of both object pose $\boldsymbol{\theta}_i$ and the geometric model $\mathcal{M}$. Vector $\boldsymbol{\eta}_i$ represents some white Gaussian noise process with covariance $\boldsymbol{N}_i$. Given a feature vector computed from the image data, the object state $\boldsymbol{x}_i$ is estimated from the predicted state $\hat{\boldsymbol{x}}_i$ by extended Kalman-filtering (EKF)

$$\boldsymbol{x}_i = \hat{\boldsymbol{x}}_i + \boldsymbol{K}_i(\boldsymbol{f}_i - \boldsymbol{H}_i \hat{\boldsymbol{x}}_i).$$

The most computation intensive part is the calculation of the Kalman-gain matrix $\boldsymbol{K}_i$ based on the predicted covariance matrix $\hat{\boldsymbol{P}}_i$.

$$\boldsymbol{K}_i = \hat{\boldsymbol{P}}_i \boldsymbol{H}_i' (\boldsymbol{H}_i \hat{\boldsymbol{P}}_i \boldsymbol{H}_i' + \boldsymbol{N}_i)^{-1} \tag{2}$$

For a feature vector of dimension n the time complexity for computing $\boldsymbol{K}_i$ is thus $\mathcal{O}(n^3)$. If in two subsequent images – for instance due to occlusion – different features are selected, the corresponding rows in matrix $\boldsymbol{H}_i$ have to be either added or discarded. Thus the measurement matrix of the Kalman filter becomes time-variant in dimension. In order to avoid a systematic distortion of the state estimate by a continuously changing measurement matrix, Gengenbach [5] suggests using an iterated extended Kalman-filter (IEKF). This, however, increases computation time for k iterations to $\mathcal{O}(kn^3)$.

The structure of the Kalman-filter becomes much simpler and hence computationally more efficent, if rather than an image feature vector $\boldsymbol{f}$ the three dimensional target pose $\boldsymbol{\theta}_m$ serves as a measurement. In this case matrix $\boldsymbol{H}_i$ reduces to

$$\boldsymbol{H}_i = [\boldsymbol{I}_{6\times6} | \boldsymbol{0}],$$

where $\boldsymbol{I}_{6\times6}$ is the 6x6 identity matrix. The matrix to be inverted in equation (2) is then 6x6 independently of the number of features selected for pose estimation. Furthermore the computation of the gain matrix, and hence the state estimate are totally independent of the image features currently selected. The reason why such an approach is usually not used for tracking [3] is that standard algorithms that can compute 3-D object pose from a single camera image (eg. [10]) are not efficient enough to reduce the total amount of computation.

The major contribution of this paper is therefore an extremely efficient model-based method for reconstructing 3-D object pose from a set of different sensor measurements which is described in the next section. Based on the 3-D target pose computed by our algorithm, a Kalman-filter simplified as described above exploits temporal redundancy for target state estimation.

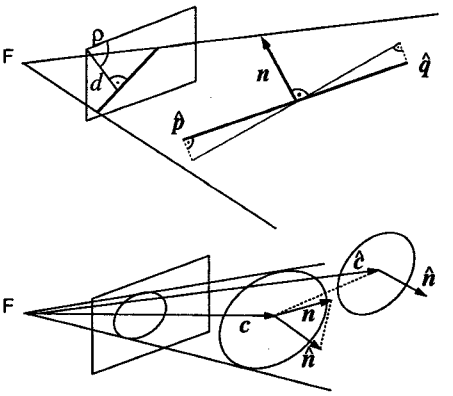

Figure 1. Top: 3-D pose error measure between a model edge $(\hat{\boldsymbol{p}}, \hat{\boldsymbol{q}})$ and an image edge (ρ, d). Bottom: 3-D pose error measure between a circle $(\boldsymbol{n}, \boldsymbol{c})$ derived from an image ellipse and a model circle $(\hat{\boldsymbol{n}}, \hat{\boldsymbol{c}})$. Dotted lines indicate the errors.

3 Efficient Reconstruction of 3-D Pose

As the pose reconstruction procedure is embedded within a Kalman-filtering process, the task is simplified by the assumption that an optimal prediction $\hat{\boldsymbol{\theta}}_i$ for the target pose to be expected in time step i is available from the evaluation of previous images. Therefore the pose reconstruction algorithm only needs to detect a – supposedly small – deviation from $\hat{\boldsymbol{\theta}}_i$. Hence the algorithm consists of three major steps: 1) feature extraction by a *local* search based on the projection of the model data from $\hat{\boldsymbol{\theta}}_i$, 2) a *partial* 3-D reconstruction of the image features, and, 3) a least-squares fit of the model shape to the reconstructed features. Note that the goal of step 2) is to eliminate the effects of camera perspective, which greatly simplifies model fitting. Considering the structures usually found in technical objects, we will focus of two types of image features, namely image lines and ellipses which are the projection of circular model features in 3-space. Furthermore, we will also take into account 3-D point measurements from range scanning sensors such as the DLR laser-scanner [7].

3.1 Error terms

The key concept for obtaining an effcient pose reconstruction procedure is to define a pose error in 3-D space rather than in the image plane. However, for many image

features, such as edge segments, it is not possible to reconstruct the corresponding 3-D edge segment. Nevertheless, for a single 2-D edge segment given by its normal orientation ρ and offset d, the normal of 3-D plane, which passes through the corresponding 3-D edge, can be computed by

$$\boldsymbol{n} = (s_x \cos\rho, s_y \sin\rho, -d)',$$

where s_x and s_y are the camera scale factors (figure 1, top). This plane is sufficient to define an error measure E^L from which we can compute by minimization a three dimensional rotation matrix $\boldsymbol{R}$ and a translation vector $\boldsymbol{t}$ that align the object model at pose $\hat{\boldsymbol{\theta}}_i$ to the current measurements.

$$E^L(\boldsymbol{R},\boldsymbol{t}) = (\boldsymbol{n}'(\boldsymbol{R}\hat{\boldsymbol{p}}+\boldsymbol{t}))^2 + (\boldsymbol{n}'(\boldsymbol{R}\hat{\boldsymbol{q}}+\boldsymbol{t}))^2 \quad (3)$$

The vectors $\hat{\boldsymbol{p}}$ and $\hat{\boldsymbol{q}}$ are the two end points of the model edge transformed to the predicted pose $\hat{\boldsymbol{\theta}}_i$ (figure 1, top). Three non-collinear image edges are necessary to compute a pose update from a single image.

Given the parameters of an image ellipse, the normal $\boldsymbol{n}$ and the center point $\boldsymbol{c}$ of the corresponding 3-D circle can be computed in closed form if the radius in 3-D space is known [2]. From the two solutions that are obtained, the correct one can be easily selected choosing the pair that is closer to the predicted model normal $\hat{\boldsymbol{n}}$ and model center $\hat{\boldsymbol{c}}$. Again, both $\boldsymbol{n}$ and $\boldsymbol{c}$ impose a spatial pose constraint on the object model from which we can derive an error term depending on $\boldsymbol{R}$ and $\boldsymbol{t}$ (figure 1, bottom).

$$E^C(\boldsymbol{R},\boldsymbol{t}) = \alpha\|\boldsymbol{n} - \boldsymbol{R}\hat{\boldsymbol{n}}\|^2 + \beta\|\boldsymbol{c} - (\boldsymbol{R}\hat{\boldsymbol{c}} - \boldsymbol{t})\|^2 \quad (4)$$

Here α and β are scalar weighting factors that account for the different noise level of $\boldsymbol{n}$ and $\boldsymbol{c}$. Note, that a single image ellipse can uniquely determine up to five degrees of freedom of 3-D pose.

Finally, one may wish to include 3-D point measurements of range scanning devices to improve accuracy by using multiple sensors (sensor fusion). Given a measured 3-D object point $\boldsymbol{p}$ we can define an error

$$E^P(\boldsymbol{R},\boldsymbol{t}) = \|\boldsymbol{p} - (\boldsymbol{R}\hat{\boldsymbol{p}} - \boldsymbol{t})\|^2. \quad (5)$$

Vector $\hat{\boldsymbol{p}}$ represents the 3-D point on the model shape at the predicted pose that corresponds to $\boldsymbol{p}$. As $\hat{\boldsymbol{p}}$ usually cannot be uniquely determined, we use the approximation suggested in [1]. Given $\boldsymbol{p}$, $\hat{\boldsymbol{p}}$ is chosen as the point on the model shape that is closest to $\boldsymbol{p}$ in terms of Euclidean distance.

3.2 Efficient Minimization

Based on the error terms defined above, the pose estimation task can be formulated as a minimization problem.

$$\min_{\boldsymbol{R},\boldsymbol{t}} \left\{ \sum_{k=1}^{L} w_k E_k^L + \sum_{k=1}^{C} v_k E_k^C + \sum_{k=1}^{P} u_k E_k^P \right\}, \quad (6)$$

where w_k, v_k and u_k are scalar weighting factors. The minimization defined in equation (6) must be subject to the constraint that matrix $\boldsymbol{R}$ be orthonormal, leading to a set of non-linear constraint equations. In order to solve for $\boldsymbol{R}$ and $\boldsymbol{t}$ an iterative minimization procedure for non-linear functions is required which is usually to complex to be used within a real-time loop. However, as we are expecting only a small deviation from the predicted object pose $\hat{\boldsymbol{\theta}}_i$, matrix $\boldsymbol{R}$ can be approximated by a *differential rotation* [11].

$$\boldsymbol{R}\boldsymbol{x} \doteq \boldsymbol{x} + \begin{pmatrix} 0 & -\delta z & \delta y \\ \delta z & 0 & -\delta x \\ -\delta y & \delta x & 0 \end{pmatrix} \boldsymbol{x} = \boldsymbol{x} + [\boldsymbol{\omega}]\boldsymbol{x}. \quad (7)$$

With this approximation the error terms in equation (6) can be linearized and a closed-form solution for $\boldsymbol{\omega} = (\delta x, \delta y, \delta z)'$ and $\boldsymbol{t}$ can be derived by purely algebraic manipulation. Two different types of error terms appear in equations (3) to (5). With the equality $[\boldsymbol{\omega}]\boldsymbol{x} = -[\boldsymbol{x}]\boldsymbol{\omega}$ the edge error term of (3) reduces to[1]

$$\sum_{k=1}^{L} (\boldsymbol{n}_k'(\boldsymbol{R}\hat{\boldsymbol{p}}_k + \boldsymbol{t}))^2 \doteq \sum_{k=1}^{L} (\boldsymbol{n}_k'(\hat{\boldsymbol{p}}_k - [\hat{\boldsymbol{p}}_k]\boldsymbol{\omega} + \boldsymbol{t}))^2$$

Expanding the right hand side, differentiating the resulting error term with respect to $\boldsymbol{\omega}$ and $\boldsymbol{t}$, and setting the derivatives equal to zero, we obtain a set of six linear equations for the unknowns.

$$\begin{bmatrix} -\sum \boldsymbol{n}_k\boldsymbol{n}_k' & \sum \boldsymbol{n}_k\boldsymbol{v}_k' \\ -\sum \boldsymbol{v}_k\boldsymbol{n}_k' & \sum \boldsymbol{v}_k\boldsymbol{v}_k' \end{bmatrix} \begin{bmatrix} \boldsymbol{t} \\ \boldsymbol{\omega} \end{bmatrix} = \begin{bmatrix} \sum (\boldsymbol{n}_k'\boldsymbol{p}_k)\boldsymbol{n}_k \\ \sum (\boldsymbol{n}_k'\boldsymbol{p}_k)\boldsymbol{v}_k \end{bmatrix} \quad (8)$$

For readabilty we have used the substitution $\boldsymbol{n}_k'[\boldsymbol{p}_k] = \boldsymbol{v}_k'$. Note that these equations are equivalent to those of the pose estimation algorithm from edge segments reported in [9].

Similarly, the Euclidean distance-based error terms in equations (4) and (5) can be linearized.

$$\sum_{k=1}^{P} \|\boldsymbol{p}_k - (\boldsymbol{R}\hat{\boldsymbol{p}}_k + \boldsymbol{t})\|^2 \doteq \sum_{k=1}^{P} \|\boldsymbol{p}_k - (\hat{\boldsymbol{p}}_k - [\hat{\boldsymbol{p}}_k]\boldsymbol{\omega} + \boldsymbol{t})\|^2$$

Again, expanding the right hand side, differentiating the resulting error term with respect to $\boldsymbol{\omega}$ and $\boldsymbol{t}$, and setting the derivatives equal to zero, leads to a system of six linear equations in $\boldsymbol{\omega}$ and $\boldsymbol{t}$.

$$\begin{bmatrix} \sum [\boldsymbol{p}_k] & -P\,\boldsymbol{I}_{3\times 3} \\ \sum [\boldsymbol{p}_k]^2 & -\sum [\boldsymbol{p}_k] \end{bmatrix} \begin{bmatrix} \boldsymbol{t} \\ \boldsymbol{\omega} \end{bmatrix} = \begin{bmatrix} \sum \boldsymbol{e}_k \\ \sum [\boldsymbol{p}_k]\boldsymbol{e}_k \end{bmatrix}, \quad (9)$$

where $\boldsymbol{e}_k = \hat{\boldsymbol{p}}_k - \boldsymbol{p}_k$. Combining equations (8) and (9) we obtain an approxiamte solution to the minimization stated in equation (6) that is of the form

$$\boldsymbol{A}\begin{bmatrix} \boldsymbol{t} \\ \boldsymbol{\omega} \end{bmatrix} = \boldsymbol{b} \quad (10)$$

[1] Due to space limitations only one endpoint of the model edge is included and the scalar weights have been omitted. The extension of the equations is, however, straightforward.

The important point is that the dimension of the system of equations to be solved is 6x6 independently of the number of features considered, and the time complexity required for the calculation of the coefficients $\boldsymbol{A}$ and $\boldsymbol{b}$ depends only *linearly* on the number of features selected.

As the resulting rotation vector $\boldsymbol{\omega}$ is based on a differential approximation, the application of a homogenuous transformation matrix derived from $\boldsymbol{\omega}$ and $\boldsymbol{t}$ to the model data will not precisely align the model to the reconstructed measurements. The iterative application of (10), however, will yield such precise alignment as the computed rotation will keep decreasing, which in turn increases the accuracy of the differential approximation. In practical tracking applications three to four iterations are usually sufficient to obtain optimal alignment.

3.3 Feature Selection

Having computed the 3-D object pose $\boldsymbol{\theta}_m$ as described above, a Kalman-filter that uses $\boldsymbol{\theta}_m$ as measurement estimates the object state $\boldsymbol{x}_i$ and calculates a prediction $\hat{\boldsymbol{\theta}}_{i+1}$ of the relative object pose to be expected in the next image. Based on this prediction, feature extraction in the next image is prepared by projecting the model shape $\mathcal{M}$ from pose $\hat{\boldsymbol{\theta}}_{i+1}$ into the image plane according to the camera model. In order to account for potential occlusions, hidden lines in the projection are eliminated by an efficient analytic hidden-line removal algorithm for general polyhedra [12]. Thus not only self-occlusion can be considered but also occlusions of the target by other objects. From the set of visible edges a subset is selected by a heuristic selection procedure, which tries to maximize 1) edge length, 2) mutual edge distance, and, 3) edge intersection angles. The last criterion is intended to ensure good numerical conditioning of 3-D pose estimation, while the other two are supposed to favor features that can be reliably extracted from the image. Based on the location of selected features a rectangular region of interest in the image is defined to which feature extraction is restricted (figure 2).

The selection of elliptic features is also straightforward. For visibilty determination a polyhedral object model is required, where circular faces are approximated by polygons. Such polygons are annotated by the circle parameters needed by the reconstruction algorithm, which are the circle normal $\boldsymbol{n}$, center $\boldsymbol{c}$ and radius r. From the projection of the polygonal approximation the visibile portions of the resulting ellipse can be determined, and appropriate regions of interest can be easily computed.

4 Experimental Results

4.1 Image Preprocessing and Feature Extraction

Both the Kalman-filter and the pose reconstruction algorithm described above are implemented on an SGI Indigo^2 workstation that is connected via shared memory to a Datacube MV200 image processor, where image acquisition, preprocessing and feature extraction is done. The preprocessing steps include image rectification to eliminate lens distortions, horizontal and vertical gradient filtering with a 7x7 kernel, and gradient magnitude computation. As the feature extraction step only needs to detect a small deviation from an expected feature position within a small region of interest, edge parameters are determined by a *Hough-transform* in an extremely limited parameter space. Thus the notorious time and space complexity of the Hough-transform can be avoided, and we can fully benefit from its well-known robustness against low contrast and partial occlusion. Four image edges with lengths up to 120 pixels can thus be tracked at video rate (25Hz), up to nine such features at half video rate. As a fast Hough-transform based approach to ellipse detection in a seperable parameter space is still under development, the results presented below were all obtained with edge features. The dynamic model encoded in the Kalman-filter is for all experiments a second-order, constant acceleration motion in each positional parameter. Presently, initialization is interactive.

4.2 Feature Selection and Estimation Accuracy

Figure 2 shows an example of dynamic feature selection. The robot is interactively moved to grasp an object that is observed by the tracking system. Apart from the object model, a rudimentary model of the gripper jaws is included. During motion, the system tracks the target and continuously predicts the set of visible object edges from which appropriate regions of interest are automatically computed. Thus accurate tracking is still possible, even if more than half of the object is occluded by the gripper.

Figure 3 shows some statistics on pose estimation accuracy. To collect the data the robot was moved on a randomly generated trajectory located over a non-moving object. At 100 randomly chosen setpoints the current estimation of the relative object pose was transformed into the robot base frame. Ideally all computed poses should be equal in base coordinates. Figure 3 shows the resulting RMS for each positional parameter. Obviously, rotations about the optical axis (τ) and translations parallel to the image plane (x and y) can be determined more accurately than depth or rotations about the axes parallel to the image plane. This is typical of monocular vision. However, the resulting errors are close to the theoretical optimum that can be achieved with the given setup (image resolution 384x287 pixels, focal length 4mm, average object distance 170mm, object diameter 40mm).

4.3 Closed-Loop Control

In order to control the robot in closed loop based on the visual measurements obtained by the tracking system,

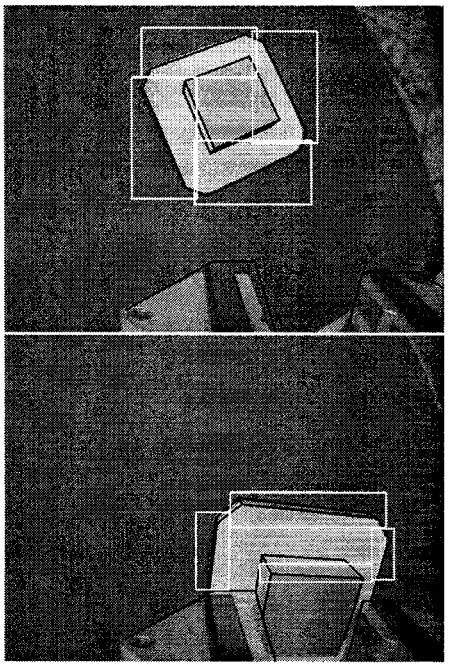

Figure 2. Dynamic handling of occlusion: The CAD model of the object is projected into the image at the pose estimated by the tracking system (black lines). Also, a simple model of the gripper is included. The regions of interest for feature extraction are drawn white. By prediction and continuous hidden-line removal visible features are automatically selected for tracking.

a Cartesian reference position with respect to the object is defined. The control error to be minimized is the Cartesian pose difference between the robot tool center point and the currently estimated reference pose. The robot control system requires an incremental Cartesian update at an 8msec rate, while the vision cycle time is 40msec. In order to match the different update rates and to compensate small deviations in computational delay (±5msec), a Cartesian trajectory generator incrementally computes robot commands based on the current pose of both robot and target. The interpolator is designed such that a smooth acceleration profile is guaranteed and both acceleration and velocity limits are fully exploited. The control performance achieved in this manner is far superior to that obtained with simple, fixed-gain PI-control.

Figure 4 shows a typical sequence of 6-dof visual control. Note that due to the robustness of the Hough-transform feature extraction is reliable and accurate despite a cluttered background and partial occlusion of tracked features by the person's hand. Figure 5 shows a plot of one translatorial pose component when tracking an object mounted on a second robot that performs a sinusoidal

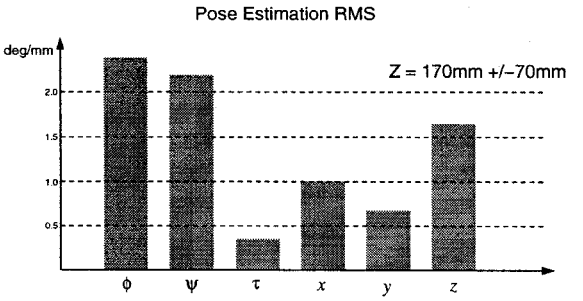

Figure 3. Pose estimation RMS error computed by tracking a non-moving object with a single camera from a random robot trajectory.

motion in several degrees of freedom. Due to temporal filtering of the Kalman-filter the predicted pose is much smoother than the mere pose reconstruction. Moreover, although the polynomial model of motion does not match the actual target motion, the predicted values faithfully follow the measurement, with only little overshoot at the maxima. In order to compensate computational delay, the robot trajectory is computed from the predicted positions. As so far no attempt has been made to include robot dynamics into the control scheme, the robot position computed from the current joint angles lags by about 80 msec (figure 5).

5 Conclusions

A new algorithm for fast model-based pose estimation from different visual measurements has been presented. If embedded into a Kalman-filter, the 3-D pose of a target can be precisely tracked in an image sequence, while occlusions are continuously detected and handled. The method has been extensively tested in a real-time position-based visual servoing system and has proven accurate, efficient and reliable. Despite full 6-dof pose and motion estimation and continuous occlusion detection, a cycle time of 80msec can be achieved when tracking up to eight edges. Image preprocessing and feature extraction account for about 95% of computation time, which can be reduced by a more efficient use of the available image processing hardware. Future work aims at including elliptical features and multiple sensors such as stereo cameras and fast range sensors.

Check `http://www.op.dlr.de/FF-DR-RS/VISION` for video clips of recent experiments.

References

[1] P. J. Besl and N. D. McKay. A method for registration of 3-D shapes. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 14(2):239–256, February 1992.

[2] M. Dhome, J. T. Lapresté, G. Rives, and M. Richetin. Spatial localization of modelled objects of revolution in monocular

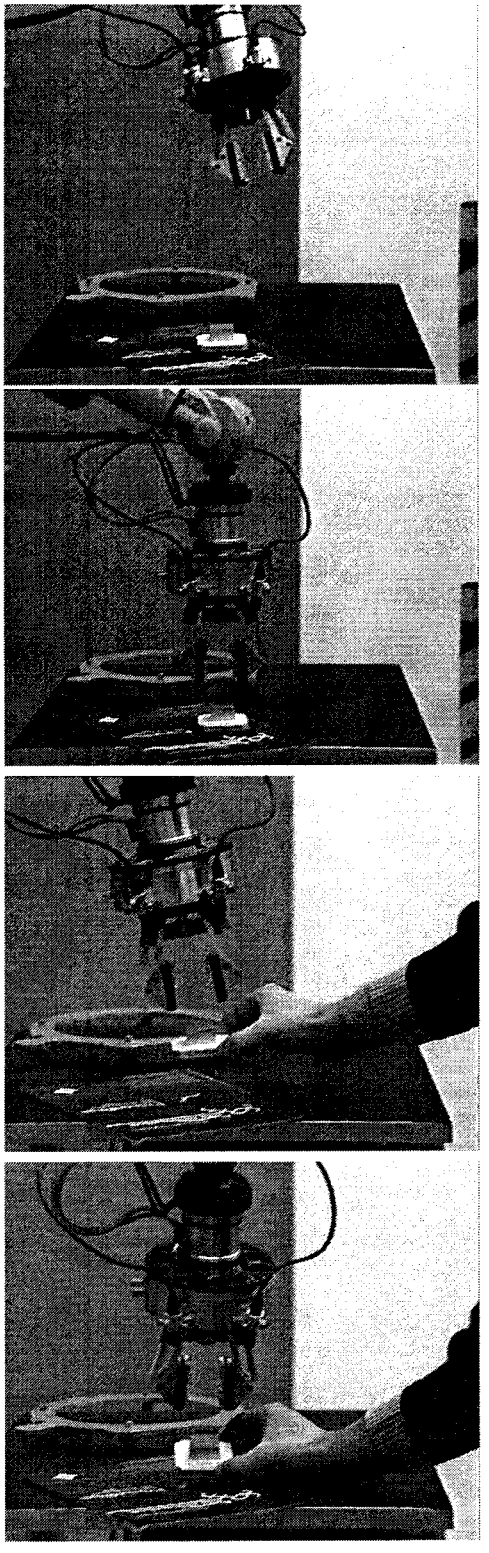

Figure 4. Align-and-Track Task: From an arbitrary starting position the end-effector is precisely servoed into a predefined reference position with respect to the target object. If the object starts moving, the robot tracks the motion in all six dof, keeping the relative reference position.

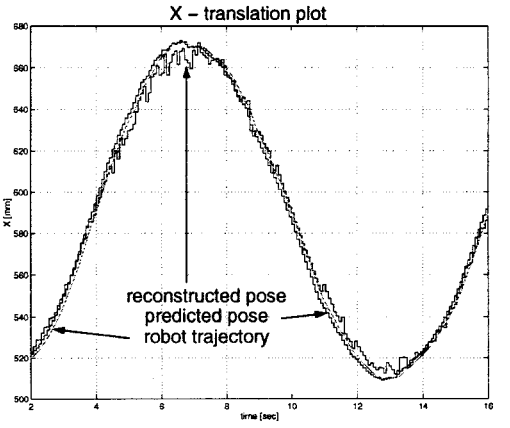

Figure 5. Results of tracking an object moving periodically in 6 dof. The input to the robot controller is computed from the Kalman-filter prediction. The dotted line shows the trajectory of the robot.

perspective vision. In *European Conf. on Computer Vision*, pages 475–485, 1990.

[3] E. D. Dickmanns and V. Graefe. Dynamic monocular machine vision. *Machine Vision and Applications*, 1:223–240, 1988.

[4] C. Fagerer, D. Dickmanns, and E. D. Dickmanns. Visual grasping with long delay time of a free floating object in orbit. *Autonnomous Robots*, 1(1):53–68, 1994.

[5] V. Gengenbach, H.-H. Nagel, M. Tonko, and K. Schäfer. Automatic dismantling integrating optical flow into a machine vision-controlled robot system. In *Proc. IEEE International Conf. on Robotics and Automation*, pages 1320–1325, April 1996.

[6] A. Graffunder and I. Hartmann. Towards manipulating moving objects I: Estimation of 3D relative movements from stereo-image sequences. In *IFIP TC 7 Conference on Modelling the Innovation*, pages 27–34, 1990.

[7] F. Hacker and J. Shi. The DLR laser scanner. Technical report, Institute of Robotics and System Dynamics. Deutsche Forschungsanstalt für Luft- und Raumfahrt - DLR, 1995.

[8] G. D. Hager, S. Hutchinson, and P. I. Corke. A tutorial on visual servo control. IEEE International Conf. on Robotics and Automation. Tutorial Notes TT3, April 1996.

[9] R. Kumar and A. R. Hanson. Robust methods for estimating pose and a sensitivity analysis. *Computer Vision, Graphics and Image Processing*, 60(3):313–342, November 1994.

[10] D. G. Lowe. Fitting parametrized three-dimensional models to images. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 60(5):441–450, May 1991.

[11] R. Paul. *Robot Manipulators*. MIT Press, Camebridge, MA, 1981.

[12] T. Peter. Effizientes Eliminieren verdeckter Kanten in polyedrischen Szenen zur Simulation von Kamerabildern. Technical Report IB-515-96-5, Institute of Robotics and System Dynamics. Deutsche Forschungsanstalt für Luft- und Raumfahrt - DLR, 1996.

[13] A. C. Sanderson and L. E. Weiss. Image-based visual servo control using relational graph error signals. *Proceedings of the IEEE*, pages 1074–1077, 1980.

[14] W. J. Wilson. Visual servo control of robots using Kalman filter estimates of robot pose relative to work-pieces. In K. Hashimoto, editor, *Visual Servoing*, pages 71–104. World Scientific, 1993.