Uncalibrated Hand-Eye Coordination with a Redundant Camera System*

Christian Scheering, Bernd Kersting
Technical Computer Science, Faculty of Technology,
University of Bielefeld, 33501 Bielefeld, Germany

Abstract

We describe a method for 3D visual manipulator control using a redundant camera system without explicit external or internal calibration. Under the assumption of a simple linear camera model a fusion equation is derived for which only three parameters have to be estimated regardless the number of cameras. In simulations as well as in real experiments the feasibility of our approach for a 3D positioning task of a six degree of freedom (DOF) Puma 200 to a target is demonstrated. It is shown that using redundant views increases positioning accuracy and fault tolerance. The achieved accuracy is sufficient to perform an additional insertion task.

1 Introduction

Using a robot manipulator for an assembly task requires the ability to grasp and insert different parts. The common approach is to solve the 3D relationship between the robot and the environment based upon a 2D vision measurements. That in turn requires the internal and external camera parameters to be calibrated which is difficult and cumbersome.

In the last years the idea of uncalibrated visual guidance found more and more attention. Skaar et al. [9] described camera space manipulation while Yoshimi and Allen [11] demonstrated 2D alignment of an eye-on-hand manipulator using rotational epipolar motion.

Both Hager [4] and Hollinghurst/Cipolla [2] exploits a nearly uncalibrated stereo camera setup. Also Hager introduced the ideas of projective invariance into the field of uncalibrated visual control [3].

Recent work shows the feasibility using the well known image Jacobian. In [6] the usefulness of adaptive differential feedback employing a visual motor Jacobian was shown. In [10] the idea of exploratory movements for dynamic image Jacobian estimation was demonstrated.

This work is a different way of uncalibrated hand-eye coordination. It is based upon a simple parallel (and therefore linear) camera model and uses several redundant arbitrary camera views in a sensor fusion approach. We show simulations and real experiments demonstrating the capability of a redundant uncalibrated camera system in order to increase position accuracy and in case of different camera failures.

2 Visual control

The key idea of our visual control is that for a Cartesian motion the image Jacobian is equivalent to the assumption of a parallel-camera model. Defining an image-based position-error and exploiting the parallel projection leads to a simple linear equation for a resulting Cartesian correction movement called the *fusion equation*. The parameters in turn are estimated with two different sensor-fusion methods – a least-square solution (LS) and in comparison a Kalman-filter approach (KF).

2.1 Model

Many researchers in the field of visual control (either with uncalibrated cameras or not) exploit the so called *image Jacobian* $\boldsymbol{J}$ introduced by Weiss e.al. [8] in order to relate a (discrete and small) displacemovement $\Delta \boldsymbol{m}$ (either in joint- or task-space) with a 2 dimensional image-feature displacement $\Delta \boldsymbol{f}$:

$$\Delta \boldsymbol{f} = \boldsymbol{J} \cdot \Delta \boldsymbol{m} \tag{1}$$

The problem is to invert the Jacobian, using a (pseudo) inversion in order to calculate the displacement $\Delta \boldsymbol{m_e}$ corresponding to an image displacement $\Delta \boldsymbol{f_e}$ defined by an appropriate feature-space error function.

*The work described in this paper has been funded by the German Research Foundation (DFG) in the project SFB 360.

We chose a different approach of how to relate a feature-space error-function with a corresponding task-space displacement. This approach is somewhat related to the image Jacobian. We use a quite rough approximation of the image-forming process – the parallel projection.

The parallel projection $\boldsymbol{P}^j$ (see [5]) of a 3D world point $\boldsymbol{m}$ in *homogeneous* coordinates $\boldsymbol{m}^w = (m_x, m_y, m_z, 1)^T = (\boldsymbol{m}, 1)^T$ onto the j^{th} camera plane is

$$\begin{aligned} \boldsymbol{f}^j &= \begin{pmatrix} r^j_{11} & r^j_{12} & r^j_{13} & t^j_1 \\ r^j_{21} & r^j_{22} & r^j_{23} & t^j_2 \end{pmatrix} \cdot \boldsymbol{m}^w \\ &= (\boldsymbol{R}^j\ \boldsymbol{t}^j) \cdot \boldsymbol{m}^w \\ &= \boldsymbol{P}^j \cdot \boldsymbol{m}^w \end{aligned} \tag{2}$$

$\boldsymbol{P}^j$ in eq. (2) is nothing but the first two rows of the corresponding homogeneous transformation ${}^{c^j}\boldsymbol{T}_w$ from the world to the j^{th} camera coordinate system.

The simplest error function for a linear point-to-point movement of a manipulator at $\boldsymbol{m}$ to a goal $\boldsymbol{g}$ is to define an appropriate error-displacement vector $\Delta\boldsymbol{d_e}$ which has to become (nearly) zero.

$$\Delta\boldsymbol{d_e} = \boldsymbol{m} - \boldsymbol{g} \to 0 \tag{3}$$

For the corresponding displacement feature $\Delta\boldsymbol{f}^j_e$ in the j^{th} camera using eq. (2) follows:

$$\begin{aligned} \Delta\boldsymbol{f}^j_e &= \boldsymbol{f}^j_m - \boldsymbol{f}^j_g \\ &= \boldsymbol{P}^j \cdot \boldsymbol{m}^w - \boldsymbol{P}^j \cdot \boldsymbol{g}^w \\ &= \boldsymbol{R}^j \cdot \boldsymbol{m} + \boldsymbol{t}^j - \boldsymbol{R}^j \cdot \boldsymbol{g} - \boldsymbol{t}^j \\ &= \boldsymbol{R}^j \cdot \Delta\boldsymbol{d_e} \end{aligned} \tag{4}$$

Eq. (4) shows additionally that the parallel projection $\boldsymbol{R}^j$ of a displacement is equivalent to the image Jacobian in eq. (1).

Given a set of three Cartesian linear independent displacement vectors[1] $\{\boldsymbol{d}_1, \boldsymbol{d}_2, \boldsymbol{d}_3\}$ the error-displacement vector $\boldsymbol{d_e}$ can be calculated by their linear combination:

$$\boldsymbol{d}_e = \sum_{i=1}^{3} \xi_i \boldsymbol{d}_i \tag{5}$$

Under the assumption of a parallel projection $\boldsymbol{R}^j$ the projected version of eq. (5) is:

$$\begin{aligned} \boldsymbol{f}^j_e &= \boldsymbol{R}^j \cdot \boldsymbol{d}_e &= \boldsymbol{R}^j \cdot \textstyle\sum_{i=1}^3 \xi_i \boldsymbol{d}_i \\ &= \textstyle\sum_{i=1}^3 \xi_i \cdot \boldsymbol{R}^j \cdot \boldsymbol{d}_i &= \textstyle\sum_{i=1}^3 \xi_i \boldsymbol{f}^j_i \end{aligned} \tag{6}$$

Hence we can define the error-function as the projection of the corresponding Cartesian displacement $\boldsymbol{d}_e$. Calculating an appropriate set of scalars ξ_1, ξ_2, ξ_3 in the image space and inserting them into eq. (5) leads directly to the desired displacement-vector in the Cartesian 3D space!

Unfortunately eq. (6) is under-determined. Therefore at least two cameras are necessary yielding an over-determined system. But on the other hand this is the way how to integrate several other camera views as well simply by solving the following over-determined system:

$$\underbrace{\begin{pmatrix} \boldsymbol{f}^1_e \\ \boldsymbol{f}^2_e \\ \vdots \\ \boldsymbol{f}^j_e \end{pmatrix}}_{\boldsymbol{z}} = \underbrace{\begin{pmatrix} \boldsymbol{f}^1_1 & \boldsymbol{f}^1_2 & \boldsymbol{f}^1_3 \\ \boldsymbol{f}^2_1 & \boldsymbol{f}^2_2 & \boldsymbol{f}^2_3 \\ \vdots & \vdots & \vdots \\ \boldsymbol{f}^j_1 & \boldsymbol{f}^j_2 & \boldsymbol{f}^j_3 \end{pmatrix}}_{\boldsymbol{H}} \cdot \underbrace{\begin{pmatrix} \xi_1 \\ \xi_2 \\ \xi_3 \end{pmatrix}}_{\boldsymbol{\xi}} \tag{7}$$

$$\boxed{\boldsymbol{z} = \boldsymbol{H} \cdot \boldsymbol{\xi}} \tag{8}$$

Eq. (8) plays the central role in our approach and is called the *fusion equation.* Only three parameters have to be estimated independently of the number of cameras and only three initial test moves are necessary (instead of several exploratory movements when performing a repeated Jacobian acquiring as in [10]).

2.2 Model solution

In the present work we used two different methods – the least-square solution and a Kalman-filter approach. The first is much simpler than the latter but the Kalman filter is known as an optimal estimator in the presence of Gaussian noise. In order to choose the appropriate method both solutions were tested in simulations as well as in real experiments.

Assuming that $\boldsymbol{H}$ is full rank (i.e $rank(\boldsymbol{H}) = min(2j, 3) = 3, j \geq 2$) a least-square solution is possible which gives a value for $\boldsymbol{\xi}$ that minimises the norm $\|\boldsymbol{z} - \boldsymbol{H} \cdot \boldsymbol{\xi}\|$:

$$\boldsymbol{\xi}_{min} = (\boldsymbol{H}^T\boldsymbol{H})^{-1}\boldsymbol{H}^T \cdot \boldsymbol{z} \tag{9}$$

When using a linear discrete Kalman filter the plant and measurement equation, assuming zero-mean, white-noise $\mathbf{v}$ and $\mathbf{w}$ are:

$$\begin{aligned} \xi(k+1) &= \xi(k) + \mathbf{v}, & \mathbf{v} \sim N(0, \mathbf{Q}) \\ \mathbf{z}(k) &= H(k) \cdot \xi(k) + \mathbf{w}, & \mathbf{w} \sim N(0, \mathbf{R}) \end{aligned} \tag{10}$$

The incremental prediction and update solutions can be found in [1]. In our approach the whole system dynamic is included in the system noise $\mathbf{v}$. The

[1] Because in the following only displacements are considered the Δ is omitted.

problem with the Kalman-filter solution is that there is no a priori information about “good” values for the covariance matrices $\mathbf{Q}$ and $\mathbf{R}$. Most of the time these values are empirically determined. Only some qualitative considerations can be made about the choice of $\mathbf{R}$. The smaller the elements of $\mathbf{R}$ are, the more the new measurements are trusted. For $\mathbf{R} \to \mathbf{0}$ the Kalman filter degenerates to a least-square solution. On the other hand larger elements of $\mathbf{R}$ should be selected, the higher the expected measurement noise is – the old measurements are trusted more and the system will tend to converge slowly.

However, the choice of $\mathbf{Q}$ is less problematic in our approach as the experiments have shown. We have chosen pure diagonal matrices for $\mathbf{Q}$, $\mathbf{R}$ and the initial state covariance $\mathbf{P}_{(\mathbf{0}|\mathbf{0})}$ with the following diagonal elements:

$$\sigma^2_{P_{(0|0)}} = 0.1, \sigma^2_Q = 0.01, \sigma^2_R = 5.0 \qquad (11)$$

The initial state-estimate is set to $\boldsymbol{\xi}_{(0|0)} = (1, 1, 1)^T$.

The algorithm for a point-to-point movement (in order to reach a docking-position, for example) is the following:

1. Select a target, make 3 Cartesian test moves and detect their image $\boldsymbol{d}_i^j$ in each camera j
2. Calculate a new $\boldsymbol{\xi}$ and a *down-scaled* correction movement $\boldsymbol{d}_c = s \cdot \boldsymbol{d}_e, 0 < s < 1$
3. Evaluate a termination criterion: if the target is reached → stop, else proceed with step 2

The explicit down-scaling of every correction movement in step 2 is done for security reasons.

3 Simulations

In the simulations the least-square solution is compared with the Kalman filter in order to chose the better one. Additionally, the system behaviour using redundant cameras and its robustness in potential failure situations is investigated. Another benefit of simulations is that in short time numerous iterations without supervising are obtainable (e.g. for the examples below several thousand runs were evaluated).

The task is to position the manipulator tool center point at a target position. The images of these points are generated using a pin hole camera model for each view. However, for the algorithm the projected points are used only and not the information of the simulated camera. This is still an idealisation since in reality there is no guarantee that the measured points in the images are the projection of the *same* 3D point. At least they should be closed neighbours.

Each measurement of the target- and manipulator-position is overlayed with 2 dimensional Gaussian noise with a variance of 5 in horizontal and vertical direction each.

In the simulation setup each test move has a 50mm length aligned with the robots coordinate system. The distance to be moved is about 375mm. Each camera has a distance of approximately 2m from the scene. The used pin-hole cameras have a uniform scaling of 70 pixel/mm and a focal length of 20mm.

In order to show that even under the assumption of a parallel projection our iterative approach still holds, the parallel camera model is *not* used to simulate the feature generation. Fig. 1 shows a performed target-approach. The first three moves are the initial test moves. At present three different termination criteria have been used:

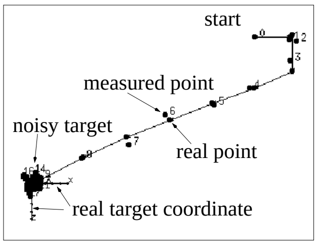


Figure 1: Simulated point-to-point move under noise.

- Maximum number of iterations $I_{max} = 200$
- Two dimensional minimal distance d_2; the approach is stopped if in *every* image the projection of the moved distance is less than d_2 pixel
- Three dimensional minimal distance d_3; the approach is stopped if the last three real motions have been less than d_3 mm each.

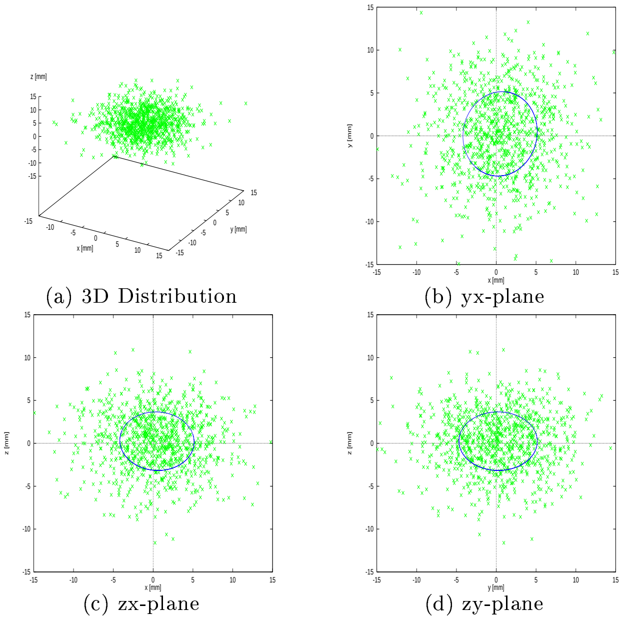
The experiments were run 1000 times each with two cameras one observing the xz-plane and the other one the yz-plane. Tab. 1 shows for both the least-square solution and the Kalman filter the results for different termination criteria. The number of runs n_r with successful termination due to the criterion, the corresponding mean number of iterations n_i and the mean 3D residual distance d after termination are displayed. For those runs which were terminated by exceeding the iteration limit the max-iteration residual d_m is shown, too.

For both termination criteria d_2 and d_3 the (trivial) observation is that the weaker the criterion, the more it fires. However, weakening the criteria does not increase the mean target-distance d significantly. The best d is achieved with the LS and d_3 = 3mm but for nearly every simulation the KF gives the minimal number of iterations. In order to have a criterion

		Criterion				
		d_2		d_3		d_2 or d_3
		2_{p}	5_{p}	3_{mm}	5_{mm}	5_{p} or 5_{mm}
LS	n_r	371	1000	166	909	1000
	n_i	97	34	106	57	29
	d/d_m	7.2/8.5	7.1	5.5/8.4	6.8/10.1	7.2
KF	n_r	380	1000	573	1000	1000
	n_i	97	32	92	28	22
	d/d_m	6.4/7.5	6.9	6.0/8.3	6.6	6.9

Table 1: Comparing the quality of LS and KF for different termination criteria.

that (nearly) always fires and yields a (nearly) minimal number of iterations and a (nearly) minimal residual distance we suggest a combined criterion, shown in the last column of Tab. 1. Although it does not produce the best residual distance it provides the minimal number of iterations. Therefore this criterion was used in the real experiment described below.



(a) 3D Distribution

(b) yx-plane

(c) zx-plane

(d) zy-plane

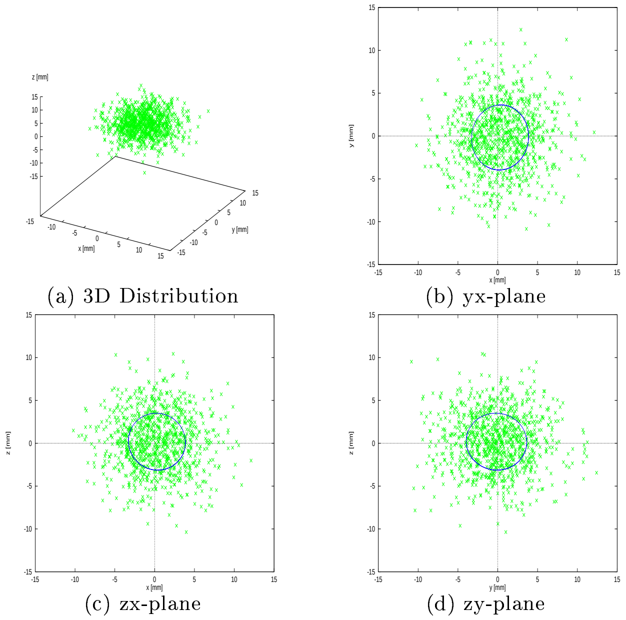
Figure 2: End position distribution using 2 cameras.

An example of the end-positions distribution using a Kalman filter with the combined criterion *5p or 5mm* is shown in Fig. 2. The target position has been transformed into the origin. Each ellipse is equivalent to the standard deviation calculated from the covariance of the appropriate distribution. The ellipse is centered at the distributions mean value and oriented along the principal axis of the distributions covariance. It can be seen in Fig. 2(c) and (d) that the distribution around the z-Axis is more compact than the distribution around the x- and y-Axis. This is due to the fact that in this simulation the z-Axis had been observed by *both* cameras.

Therefore we should expect better results (i.e. less iterations, less d and more compact distributions) if a *redundant* third camera is introduced observing the xy-plane. This is shown in Fig. 3 using the same combined termination criterion. The densier distribution is obvious – the deviation ellipses are nearly circles and became smaller. Comparing the results for n_i and d for 1000 runs (as shown in Tab. 2) it can be seen that both the mean residual distance and the mean number of iterations decreases for d_3 and the combined criterion.

		Criterion				
		d_2		d_3		d_2 or d_3
		2_{p}	5_{p}	3_{mm}	5_{mm}	5_{p} or 5_{mm}
n_i	2 Cameras	161	32	138	28	22
	3 Cameras	196	84	78	19	19
	Diff[%]	+22	+163	-43	-32	-14
d[mm]	2 Cameras	7.1	6.9	7.0	6.6	6.9
	3 Cameras	6.0	5.4	5.0	5.6	5.7
	Diff[%]	-15	-22	-29	-15	-17

Table 2: Comparing n_i and d for a KF solution between 2 and 3 Cameras.



(a) 3D Distribution

(b) yx-plane

(c) zx-plane

(d) zy-plane

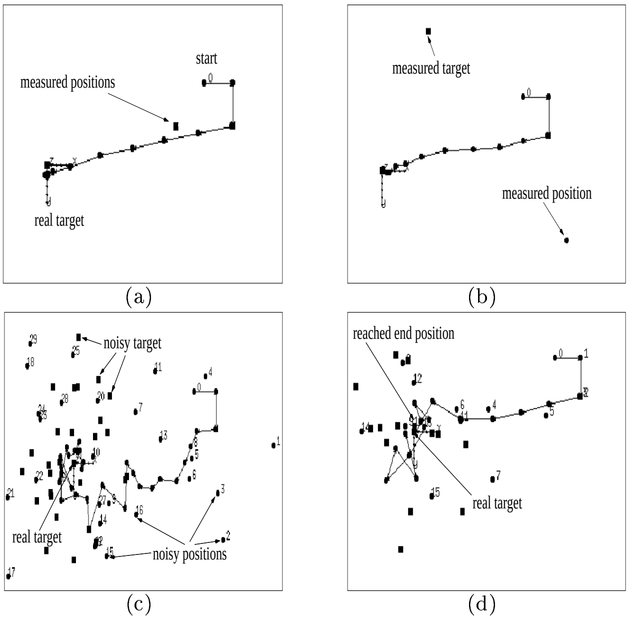
Figure 3: End position distribution using 3 cameras.

3.1 Defect simulation

Three different failure types of a single camera in a set of three have been simulated. The first is that both the target and the manipulator have always the same position. In this situation no residual information from this camera is obtained but the target is reached (see Fig. 4(a)).

The second failure is that target and manipulator have always the same but different positions. The residual is always the same and non-zero but the target is reached, too (see Fig. 4(b)).

In the last case both the target and manipulator position are very noisy. The problem is that the (very important) test moves are detected with heavy noise, too. The worse they are detected, the worse the positioning is (see Fig. 4(c)). If the test moves are detected without or with less noise (e.g. by a repetition of every move and calculating the mean) the result is improved (the target is reached after 16 iterations, see Fig. 4(d)).



(a) (b) (c) (d)

Figure 4: Different failure situations.

In order to increase the robustness of the system in the case of a camera defect, these results suggest the use of redundant cameras which are fairly easy to incorporate in our approach.

4 Experiment

In this section we demonstrate the quality of our approach in a real experiment. The manipulator is mounted on a 6 DOF Puma 200 using RCCL [7] as the control language. According to the simulation results the real experiment is shown only for the better performing Kalman-filter method.

The cameras are in approximately 1.5m distance. The target is a hole with radius 8mm in a wooden toy cube. The manipulator carries another cube with a peg which has to be inserted. The center of the hole is at $(-30, 340, 163)$mm and the manipulator is at $(150, 250, 0)$mm. Four test series containing 64 runs have been performed using a combined termination criterion $d_2 = 2$p and $d_3 = 1$mm.

Before running a test both the target and the manipulator point to be tracked are marked by the user in each image. The tracking is performed using a template matching. Due to the selection procedure and the different perspectives of each camera the template centers are not the projection of the *same* point in 3D space.

Criterion	d[mm]	
	without	with new tests
2p	3.6	2.0
5p	3.7	0.6
3mm	3.6	0.8
5mm	3.7	0.6
5p/5mm	4.1	0.9

Table 3: Simulation of end position residual using new test moves after 9 iterations.

series	d	n_i
1	2.5	12
2	2.5	11
3	2.5	9
4	3.0	9
mean	2.6	10

Table 4: Mean real end position residual and iterations.

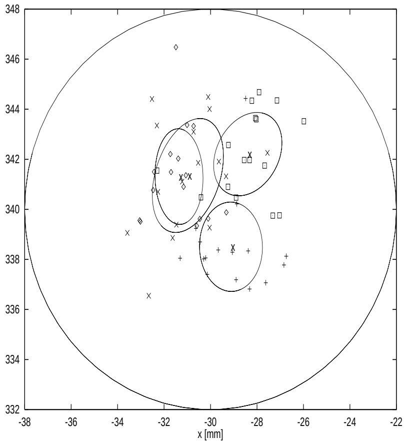
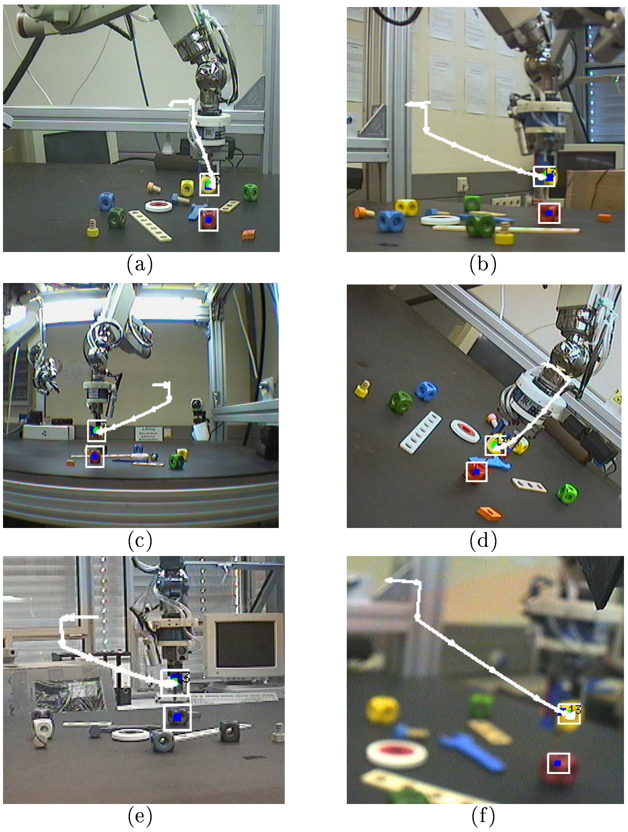


Figure 5: End point distribution in real experiment. Each ellipse around the mean value is equivalent to the distance standard deviation of a test series.

For security reasons a point above the selected target describes the desired target position. For our setup this relative correction vector is $\Delta \boldsymbol{c} = (0, 0, -50)$mm. This relative distance is projected onto each image using the parallel-camera model in eq. (4). The six parameters are calculated based on the measured projection of the test moves. This induces another error source because the parallel projection (as the equivalent image Jacobian) is only a good approximation in a small area around the measured point. It is shown in Tab. 3 that only one additional set of test moves near the desired target reduces this error.

Despite these errors (noise, parallel projection, manual target selection) the results shown in Tab. 4 for the mean target residual distance d and the mean iteration number n_i are satisfying. The hole was found in all runs and the mean distance is approximately 2.6mm from the center of the hole. Fig. 5 shows the corresponding distribution of end positions of all 64 runs. With the achieved accuracy the peg was inserted successfully simply by moving downward with a force-guarded motion.

The last series of images in Fig. 6 demonstrates the ability of our approach to *fuse* several arbitrary positioned camera views even if some images have poor quality due to high lens distortion (Fig. 6(c)) or blur (Fig. 6(f)).


(a)

(b)

(c)

(d)

(e)

(f)

Figure 6: End positions in six arbitrary views each showing the projected trajectory.

5 Conclusions

This work presented an uncalibrated visual manipulator control using redundant cameras. A parallel-camera model is used to calculate a correction. Instead of exploiting the Jacobian directly a linear combination of three linear independent test movements is performed. Independently of the number of cameras only three parameters have to be estimated using sensor-fusion methods. The quality of this approach is shown in simulations and real experiments.

The next step in this framework is to incorporate an automatic motion detection and tracking ability. Another point is to apply the known robot motion in order to estimate the pin-hole camera parameters without any further knowledge. Using this model an estimate of the epipolar geometry might be usefull in order to detect a target which has been selected in only one view. Controlling the orientation as well will be examined using additional track points on both the target and manipulator.

More work will be on parallelising – each view could be processed on a single computer and the result is fused together with the Kalman filter to reduce the overall computation time.

References

[1] Y. Bar-Shalom and X. Li. *Estimation and Tracking.* Artech House, 1993.

[2] R. Cipolla and N. Hollinghurst. Uncalibrated stereo hand-eye coordination. *Image and vision computing*, 12(3):187–, 1994.

[3] G. Hager. Calibration-free visual control using projective invariance. *Proc. Int. Conf. Computer Vision*, pages 1009–1015, 1995.

[4] G. Hager, W. Chang, and A.S. Morse. Robot feedback control based on stereo vision: Towards calibration-free hand-eye coordination. *IEEE Control Systems Magazine*, 15(1):30–39, 1995.

[5] D. Harris. *Computer graphics and applications.* Chapman and Hall, 1984.

[6] M. Jägersand, O. Fuentes, and R. Nelson. Experimental evaluation of uncalibrated visual servoing for precision manipulation. In *Proc. IEEE Int. Conf. Robot. Automat.*, pages 2874–, 1997.

[7] J. Lloyd. *RCCL User's Guide.* Computer Vision and Robotics Laboratory, 1988.

[8] A. C. Sanderson, L. E. Weiss, and C. P. Neumann. Dynamic sensor-based control of robots with visual feedback. *IEEE Trans. Robot. Automat.*, RA-3:404–417, Oct. 1987.

[9] S. Skaar, W. Brockman, and W. Jang. Camera-space manipulation. *Int. Jour. Robot. Research*, 6(4):20–32, 1987.

[10] H. Sutanto, R. Sharma, and V. Varma. Image based autodocking without calibration. In *Proc. IEEE Int. Conf. Robot. Automat.*, pages 974–, 1997.

[11] B. Yoshimi and P. Allen. Alignment using an uncalibrated camera system. *IEEE Trans. Robot. Automat.*, 12(5):516–521, 1996.