

A CAD Tool for Remote Calibration of Space Station Robotic Test Bed

Salvatore Illiano Giovanni Iodice Bruno Siciliano

Dipartimento di Informatica e Sistemistica
Università degli Studi di Napoli Federico II
Via Claudio 21, 80125 Napoli, Italy

Abstract — This paper describes the evolution of a CAD programming tool for remote video calibration of the Internal Automation & Robotics Technology Test Bed (IARTTB) available in the WKR Section at ESTEC. First the correspondence between the CAD model of the workcell in the off-line programming system and the real world is created. Then the robot is remotely calibrated with respect to the two racks present in the workcell by using a video camera on the end effector. A software package with a user-friendly interface has been developed to carry out the calibration procedure in a systematic way.

I. Introduction

In the latest years the interest towards space robotics applications is increasing, thanks to the foreseen construction of space stations. To carry out duties in internal and external operations it may become convenient to use robot manipulators, especially if those have to work in a known structured environment.

The launch stress and altered gravity conditions may modify some physical parameters of the manipulator, which then requires a calibration procedure to find an accurate relationship between the joint variables and the end-effector location to be used by the controller in planning and execution tasks [1],[2],[3].

Generally the calibration operations, needing the presence of an expert operator close to the robot, are not always possible (unmanned missions) or they have drawbacks such as increase of crew workload, cost, and robot utilization time.

This study investigates the possibility of performing local robot calibration remotely via the development of a standard working procedure. The approach is to compute for each location of a manipulation task the displacement error vector between the real and calculated locations of the robot end effector, and to use this vector to get an accurate execution of the task. The calibration method relies on the use of a video camera mounted on the robot end effector, and the prior knowledge-based algorithm [4] is adopted to recognize three-dimensional objects from single two-dimensional images.

The method has been implemented on the working environment available in the WKR Robotics Section of the European Space Agency establishment ESTEC. The workcell, called Internal Automation & Robotics Technology Test Bed (IARTTB), has

been conceived to develop technological solutions to the problems related to the use of a robot for Intra-Vehicular Activities (IVA) servicing in a space station.

To generate application programs which are not affected by manipulator positioning errors a unique correspondence between the CAD model of the workcell and the real world has to be created. The calibration of the off-line programming system is performed in two steps. In the first step three-dimensional measures of all the objects in the workcell are obtained to derive representative CAD models. In the second step the best values of the parameters are computed to update the models in the off-line programming system in order to perform an accurate inverse kinematics procedure.

A software package has been developed to handle communication between the robot controller and a workstation, to integrate the video images of the camera with those of the CAD system, and to compute the displacement between the actual and the expected locations of the robot end effector. A user-friendly interface has been realized to simplify and speed up the calibration procedure.

The performance of the calibration system is demonstrated by means of numerical examples on the above robotic test bed.

II. Video Calibration

The video calibration procedure has been implemented by using the prior knowledge-based approach in [4] which is effective when the vision system operates in a completely structured environment, i.e. when the types of objects are known and it is desired to compute their location in space.

The following equations describe the projection of a 3D object model point p into a 2D image point of coordinates (u, v) :

$$p' = Rp \quad (1)$$

$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} \frac{f_u p'_x}{(p'_z + d_z)} + d_x \\ \frac{f_v p'_y}{(p'_z + d_z)} + d_y \end{bmatrix} \quad (2)$$

where R is the rotation matrix transforming p in the original object frame into a point $p' = (p'_x, p'_y, p'_z)$ in the camera frame. These coordinates are then combined by the parameters f_u, f_v which are proportional to the camera focal lengths along the two axes of the image plane to perform the perspective projection into a point of coordinates (u, v) ; the parameters d_x and d_y specify the coordinates of the origin of the object frame on the image plane whereas d_z represents the distance between the origins of the two frames.

The rotation matrix R can be expressed as the initial rotation matrix of the object with respect to the camera frame premultiplied by three rotation matrices describing incremental rotations by (ϕ_x, ϕ_y, ϕ_z) about the axes of the current frame; such rotations are to be considered nearly independent for small increments.

Therefore the goal is to solve (1),(2) for $h = [d_x \ d_y \ d_z \ \phi_x \ \phi_y \ \phi_z]^T$, given a number of object model points and their corresponding locations into an image available

by the camera. This can be achieved by applying Newton's method which requires computation of partial derivatives of u and v with respect to the unknown parameters. It is not difficult to show that:

$$\frac{\partial u}{\partial d_x} = 1 \quad \frac{\partial u}{\partial d_y} = 0 \quad \frac{\partial u}{\partial d_z} = -f_u c^2 p'_x \quad (3)$$

$$\frac{\partial u}{\partial \phi_x} = -f_u c^2 p'_x p'_y \quad \frac{\partial u}{\partial \phi_y} = f_u c(p'_z + c p_x'^2) \quad \frac{\partial u}{\partial \phi_z} = -f_u c p'_y \quad (4)$$

and

$$\frac{\partial v}{\partial d_x} = 0 \quad \frac{\partial v}{\partial d_y} = 1 \quad \frac{\partial v}{\partial d_z} = -f_v c^2 p'_x \quad (5)$$

$$\frac{\partial v}{\partial \phi_x} = -f_v c(p'_z + c p_y'^2) \quad \frac{\partial v}{\partial \phi_y} = f_v c^2 p'_x p'_y \quad \frac{\partial v}{\partial \phi_z} = f_v c p'_x \quad (6)$$

where $c = 1/(p_x'^2 + d_z)$.

For each point in the model that should match against some corresponding point in the image, the model point is projected into the image using the current parameter estimates and the errors e_u and e_v are measured with respect to the given image point. Then the *two* equations:

$$\frac{\partial u}{\partial h} \Delta h = e_u \quad \frac{\partial v}{\partial h} \Delta h = e_v \quad (7)$$

are used for *three* different points to form a linear system of *six* equations to be solved for the vector of corrections Δh at each iteration. If the *two* focal lengths are not exactly known, they can be easily embedded into the algorithm by computing the partial derivatives of u and v with respect to f_u and f_v ; in that case a *fourth* point is needed to generate the corrections Δf_u and Δf_v at each iteration.

III. Space Station Robotic Test Bed

The Internal Automation & Robotics Technology Test Bed (IARTTB) is a technology workcell aimed at developing solutions to the problems related to the use of a robot for Intra-Vehicular Activities (IVA) servicing in a space station. The test bed is composed of:

- A 6-DOF industrial manipulator COMAU SMART 10, equipped with a passive Remote Center of Compliance (RCC), a tool exchange device that can handle a two-finger gripper Shunk SEG 10 or a screwdriver. Special brackets, fixed to the gripper, allow placing of two video cameras on it.
- Two racks, equipped with several objects and their corresponding storage locations, one drawer and two locations to host the two robot tools.

All the objects or moving parts in the two racks are equipped with special fixtures designed to optimize the contact with the gripper fingers during the grappling operations. The robot may perform the classical operations of picking objects from one rack and placing them in the other rack so as to simulate tasks to be executed automatically in a space station.

The robot is controlled by the COMAU C3G controller and an Off-line Programming System (OPS) is available for programming, operating and monitoring the robot in an easy and fast way. The OPS is composed of a Silicon Graphics IRIS workstation with a CAD system especially developed by Tecnomatix for robotics applications, named RobCAD. This software package includes the robot model and allows designing a complete model of the workcell and of the grasping fixture (Fig. 1) as well as planning and simulating tasks by simple mouse operations. Once task feasibility has been ascertained, the corresponding procedure is implemented automatically in PDL2 (Program Definition Language) on the controller.

IV. CAD Tool

The calibration procedure normally used to calibrate the OPS consists in:

- measuring the robot workcell and updating the CAD models inside the OPS,
- calculating extended robot parameters and creating an accurate inverse kinematics solution inside the OPS.

Whereas the first step does not present any technical difficulties, the second one requires several extensive and accurate measurements which have to be performed by an expert operator using special tools, such as theodolites. This is not useful for a test bed conceived for IVA, because the time required to perform the required measurements makes inefficient use of the manipulator in a space environment.

An alternative approach is based on the video calibration technique previously described. Once the camera location is known with high accuracy, the end-effector location can be computed and calibration of the OPS becomes simple, systematic and remotely executable. The main features of the CAD tool developed to perform the calibration procedure are described in the following.

While the robot model is included in RobCAD, the model of the workcell has to be created by the user. This operation can be achieved in three steps:

- measurement of the real location for each object in the workcell,
- design of their CAD models,
- graphic construction of the workcell model.

The measurements of real object locations were performed by ESTEC Metrology Office using the 3D theodolite-based measuring system WILD TMS with a measuring accuracy of ± 0.1 mm; all measurements were expressed in the robot base frame.

The design of each component of the workcell has been implemented in a special environment of RobCAD, named *component*, by referring to the original drawings used for their construction.

The graphic assembly has been carried out in a different environment of RobCAD, named *workcell*. The accuracy in the location of the rack models has the same order as that reached in the measurement process.

During the implementation of the automatic video calibration procedure it is necessary to build a list of correspondences between the locations and the associated correction vectors to be used by the robot controller. This can be achieved according to the following steps:

- encoder measurements of real joint variables and their storage in the controller memory,
- serial communication between the controller and the workstation,
- graphic animation of the task executed by the manipulator with the transmitted data.

The first step is performed by a PDL2 procedure named **measure**. The serial communication is suitably synchronized by a C procedure which cyclically sends a new data request and recognizes a possible string of joint values transmitted from the controller. The last step is carried out by a screen refresh after each set of data which, in view of the typically slow operational speeds in space-oriented robotics applications, allows continuous animation of the CAD model.

The implementation of the video calibration procedure has been developed by devising a simple user-friendly interface. It comprises a command window for the routines to activate and three graphic windows: the first one displays the real object image shot by the camera attached to the gripper, the second one displays the virtual image of the same object obtained by the CAD model of the camera, and the third one displays the CAD model of the whole workcell.

The command window structure is realized by using the RobCAD User Interface Language, whereas a suitably modified version of the software package VideoLab has been used to show the real camera shots. To minimize errors due to image distortion, which are not taken into account in the projection equations used in the calibration procedure, only the central part of the frame is shown. The possibility that the operator selects a point on the real image that is slightly different from its corresponding point on the CAD image has been reduced by implementing a cross-shaped cursor which allows selection of the single pixel on the screen.

The solution of the linear system equations required at each step of the calibration technique is performed by a Gaussian elimination routine developed in PASCAL to transform the coefficient matrix in triangular form.

The RobCAD communication system with the database is a serial pipe and does not allow communication with other processes running outside the RobCAD environment. The linear system solving routine has to receive 2D coordinates from VideoLab and 3D coordinates from the RobCAD database. To overcome the above drawback VideoLab could have been treated as a child process launched inside the RobCAD environment from the parent calibration procedure. In this way both processes could have had access to the pipe communication system, transferring data from one to the other through the database. Unfortunately the child process does not inherit the graphic features from the parent process, so that no real image could have been shown and thus no selection of 2D coordinates would have been possible.

The interprocessing communication has been efficiently implemented by creating a shared memory segment in the RobCAD environment where all the 2D coordinates of the selected image points are stored in order to be used by the calibration procedure.

The main menu displayed by RobCAD has four main procedures: **programming**, **simulation**, **calibration**, **execution**. The last two procedures are of concern for the actual calibration procedure and their operation is described below.

To start off calibration, take the robot to the desired location by driving the joint variables obtained by kinematic inversion; an error displacement will result in view of model uncertainties.

Activate the procedure `calibration`; a new menu appears on the command window and, at the same time, the procedure `monitoring` drives the CAD model of the manipulator to the corresponding location. A shared memory segment in the RobCAD environment is also initialized so that VideoLab can access it and get the data used to form the video image.

The user can now select on the command window the function `target object`; a form will appear on the screen and the user has to select a workcell component. After this, two functions are selectable: `point` and `accept`. Clicking on the first one causes a new form to be displayed. The user has to select a point on the object model and its coordinates are obtained from the RobCAD database; then the same point has to be selected on the video image to get its 2D coordinates to be stored in the shared memory through VideoLab. If there is a mistake in the selection, the user can repeat the operation until the correct correspondence is created. Once the selections are correct, the second function has to be clicked to get the 3D coordinates which are stored as well in the assigned shared memory segment. This sequence of operations is repeated as many times as required by the calibration technique. Three couple of points are to be selected if the focal lengths are known, while four are needed if they have to be computed too.

When the point acquisition phase is over, the function `calibrate` becomes active on the command window. All the steps in the Newton's algorithm are executed, including proper transformation of 3D data from the base frame to the object frame; the initial data of h is the current one available in the RobCAD database.

Monitoring of the calibration task can be obtained by selecting the function `execution` in the main menu of the command window. A new menu appears with two functions. The first one, named `monitoring`, causes the CAD model to show the movements and the operations executed by the manipulator in real time. On the other hand the second one, named `live` shows the video camera shots generated by VideoLab.

V. Results

The video calibration procedure acquires from the RobCAD database the 3D coordinates of a set of three points chosen on the grappling fixture referred to the robot base frame and from VideoLab the corresponding 2D coordinates referred to the image plane. The obtained values are compared and the Newton's algorithm is used to obtain matching of the computed and real values of video camera location.

To simplify the debugging of the procedure, data acquisition from VideoLab has been simulated first. It is assumed that the real camera is placed in a number of different locations and the projection equations (1),(2) are used to compute the corresponding 2D points. The resulting errors between the real camera locations and the locations computed through the camera model are used to apply the calibration procedure. The results are reported in Tab. 1 for a number of 10 tests, where the two rows for each

test are respectively the initial and final displacements referred to the camera model location used as starting value by the procedure. It can be recognized that both the position and orientation final errors are very limited.

Other tests have been performed by considering errors also on the focal lengths and thus including other two points. The results, not reported here for brevity, have demonstrated that the procedure is able to estimate the focal values with a precision of the order of 10^{-4} , but the accuracy on the location parameters is decreased by an order of magnitude on the average. Also the starting displacement error had to be less than 40 mm in translation and 15 deg in rotation, otherwise the procedure did not converge [5]. From a physical viewpoint this means that the chosen points have to be projected on the linear part of the image where the distortion effect can be considered negligible.

Having checked the numerical robustness of the calibration procedure, the real camera data can be used from the various image shots. At this stage it is crucial that the vision system itself be accurately calibrated (focal lengths and origin of the image plane) because this will affect the entire robot calibration procedure. The difficulties concerned with this kind of calibration are discussed in [5].

Finally a local calibration test has been carried out. Usually the displacement between the actual robot end effector location and that assumed by the model is on the order of a few millimeters in translation and a few degrees in rotation. Therefore the accuracy requested after calibration is typically of 0.1 mm in translation and 0.1 deg in rotation.

The real manipulator end effector location measured with a 3D tool was (1146.24, -428.85, 1367.70, 0.0, 0.0, 0.2), while the result of the calibration gave (1145.87, -429.92, 1367.34, 0.64, -0.03, 0.0), that is a final displacement error of (0.37 1.07 0.36 -0.64 0.03 0.2). The obtained accuracy is less than expected and this was imputable mainly to imperfections of the calibration tool; more details can be found in [5]. Current work is in progress at ESTEC to develop a more accurate calibration tool for the vision system so as to obtain a fully satisfactory accuracy of the overall robotic test bed.

Acknowledgements

This work was performed during a six-month stay of the first two authors in the WKR Section of ESTEC, Noordwijk, The Netherlands, under the direction of Wim De Peuter. Useful tutoring by Gianfranco Visentin is gratefully acknowledged.

References

- [1] Z.S. Roth, B.W. Moring, B. Ravani, "An overview of robot calibration," *IEEE J. of Robotics and Automation*, 1987.
- [2] S. Hayati, K. Tso, G. Roston, "Robot geometry calibration," *IEEE J. of Robotics and Automation*, 1988.
- [3] J.M. Hollerbach, "A survey of kinematics calibration," *The Robotics Review*, 1, MIT Press, 1989.
- [4] D.C. Lowe, "Three-dimensional object recognition from single two-dimensional images," *Artificial Intelligence*, 31, Elsevier Science Publishers, 1987.
- [5] G. Iodice, S. Illiano, *Video Calibration of Internal A&B Technology Test Bed IARTTB*, ESTEC Working Paper n. 1721, 1993.

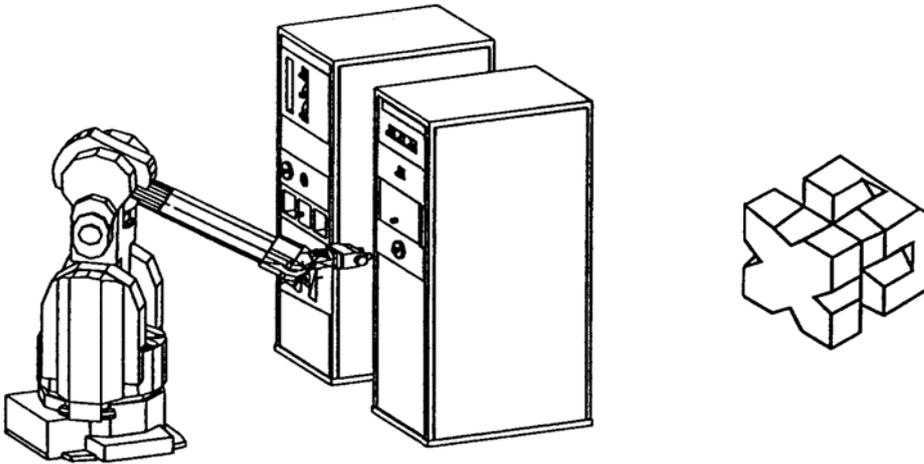


Fig. 1: CAD models of the IARTTB workcell and of the grasping fixture

<i>Test</i>	d_x [mm]	d_y [mm]	d_z [mm]	ϕ_x [deg]	ϕ_y [deg]	ϕ_z [deg]
#1	10	0	0	0	0	0
	$-1.87e-5$	$5.85e-7$	$6.99e-6$	$1.0e-7$	$1.0e-7$	$-2.0e-6$
#2	0	10	0	0	0	0
	$-1.85e-5$	$-6.69e-7$	$7.38e-6$	$1.0e-7$	$1.0e-6$	$1.0e-7$
#3	0	0	10	0	0	0
	$-1.25e-5$	$3.71e-7$	$4.10e-6$	$2.0e-6$	$1.0e-7$	$-4.0e-6$
#4	10	10	10	0	0	0
	$-4.90e-6$	$-1.78e-7$	$1.53e-6$	$1.0e-7$	$1.0e-6$	$1.0e-7$
#5	0	0	0	15	0	0
	$3.92e-6$	$-2.93e-7$	$-1.06e-6$	$1.0e-7$	$1.0e-7$	$-1.0e-6$
#6	0	0	0	0	15	0
	$1.84e-5$	$-4.99e-7$	$-1.72e-6$	$-1.0e-6$	$1.0e-6$	$1.0e-7$
#7	0	0	0	0	0	15
	$-1.94e-5$	$5.63e-6$	$7.13e-6$	$1.0e-7$	$1.0e-6$	$-1.0e-6$
#8	0	0	0	5	10	15
	$1.43e-6$	$-3.36e-7$	$-3.86e-7$	$-5.0e-6$	$-5.0e-6$	$-5.0e-6$
#9	10	10	10	5	10	15
	$-5.51e-6$	$1.34e-6$	$6.79e-7$	$-3.0e-6$	$2.0e-6$	$3.0e-6$
#10	100	100	100	5	10	15
	$-9.95e-5$	$-1.94e-5$	$-3.89e-5$	$-2.3e-5$	$7.0e-6$	$1.6e-5$

Tab. 1: Simulation results of video calibration