Programming by demonstration using fiducial markers

Deom Nathan[1] , Olivier Brüls[1] and Thierry Jacques[2]
[1]Department of Aerospace and Mechanical Engineering, Univeristy of Liège, Belgium
[2]Citius Engineering, Belgium
{npdeom,o.bruls}@uliege.be

***Abstract*— The programming by demonstration is a method that allows one to register a trajectory and to reproduce it by a robot. It could be used to speed up the programming of robots using the practical skills of workers without the time consuming code development part of classical robotic applications. This article exposes a methodology based on fiducial markers and only one classical camera which is an inexpensive solution. The fiducial markers are based on binary symbols generally used for augmented reality applications. It also presents the experimental setup created at the laboratory of human motion analysis and used to analyze the repeatability and the precision of this solution. The primary results show a promising technology and following developments will be done.**

I. INTRODUCTION

Programming by demonstration is a research field that aims to transfer the human skills to robots. Instead of programming, the end-users teach the desired behaviour to the robot. This method is composed of two phases: the teaching and the reproduction. During the teaching phase, the user shows the desired action to the robot. Afterwards, the robot programming can be automatically determined. This methodology allows one decreasing the programming time and taking advantage of the technical and adaptive skills of the end-user.

The teaching phase requires the registration of the motion. Several methods can be used [1]. For a collaborative robot, hand-guiding, which is also named lead-through programming, can be considered. However, for a classical industrial robots, the interactions with the human are restricted so other techniques should be used, such as human motion analysis methods. The measurement system could be

- a mechanical system with position encoders [5],
- a magnetic system using triangularization (e.g., ABB Simplified Robot Programming),
- an inertial system using inertial measurement units [4], [7],
- an optical system which uses cameras.

The optical systems are generally based on markers tracking, however, with the recent increase in computation power and artificial intelligence, some markerless methods also appear.

Once the teaching phase completed, the raw data should be processed to generate the robot programming. The processing could be more or less advanced depending on the needs of the final application. If required, the trajectory should be closely continuously followed as in welding [9], painting [8] or teleoperation [7]. In other cases, like pick-and-place, it is sufficient to reproduce the general behaviour [10]. Consequently, the starting point and the end point are the only interesting information of the recording.

The fiducial markers allow camera pose estimation and are generally used for augmented reality and robot localization. Several fiducial marker systems have been proposed, [2], [3]. They differ by the generation method of the marker codes but the camera pose estimation are similar, using perspective by n points problem described below. The pose (translation and rotation) of a marker is computed in the camera frame.

II. METHOD

The programming by demonstration for painting or welding application requires the tool pose measurement during the whole operation. In this work, the proposed technique is based on the computation of the tool pose on which fiducial markers are fixed using only one camera.

A. *Fiducial markers used : ArUco*

The fiducial markers selected for the project come from the ArUco library, since it is implemented in OpenCV, Open Source Computer Vision [11], which is an open source library for image and video analysis. Consequently, the ArUco library offers image processing algorithms to detect and identify the markers.

The ArUco markers are square fiducial markers composed of a binary matrix (white and black) and a black border, as it can be seen in Fig. 1. A set of markers composes a dictionary which is defined by the size of the markers sides and the number of markers. The matrices of each marker are different since they represent binary codifications selected to maximize variations between the markers, in order to easily identify the marker.

To compute the position of one marker in the camera frame, the process requires the camera parameters (distortion coefficients and camera matrix), the 2D positions of the four corners in the image and the 3D positions of the corner in the marker frame. Using these values, the Perspective by n Points (PnP) problem can be solved with n equal four which has theoretically a unique solution if the points are co-planar [12].

In a similar way, if several markers are placed on the tool, the pose of this one can be obtained using the 3D positions of the corners in the tool frame and the 2D positions of corners obtained from visible markers. The usage of more

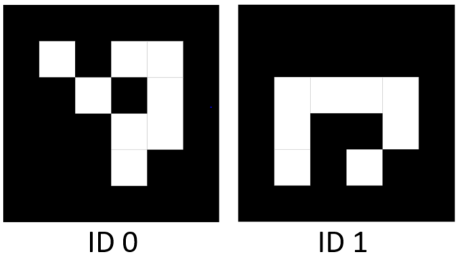


Fig. 1: Examples of the ArUco markers using the predefined dictionary DICT_4x4_50, corresponding the fifty binary codified matrix of four by four

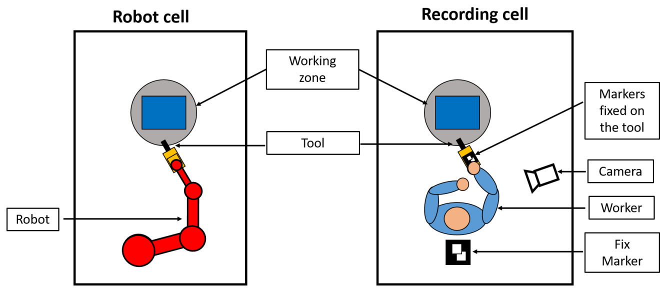


Fig. 3: Comparison between the recording cell and the robot cell

than one marker permits the detection with more orientations, since the pose can be estimated as soon as one marker is detected. The markers used to solve the problem are the one that are identified using the binary code. Then appropriate 3D positions can be used for solving the PnP problem.

The flowchart shown in Fig. 2 present the methodology used to compute the tool pose in the camera frame. The possible markers are extracted from the image. They are represented by the four corners positions. After the detection, the identification step provides the corresponding number of a marker in a specific dictionary. Using the model of the tool, the 3D positions corresponding to the corners seen are selected. The PnP problem can then be solved to obtain an estimation of the tool pose.

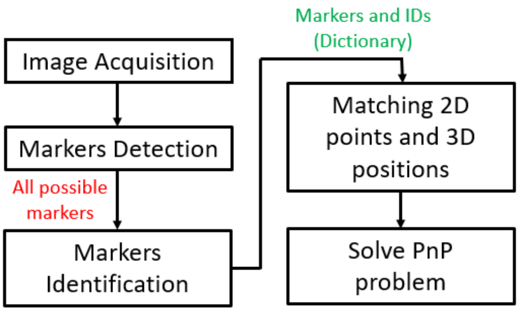


Fig. 2: Flowchart to compute the position of the tools in the camera frame, using several ArUco marker defined by a dictionary

B. Motion analysis and robot reproduction

In a programming by demonstration application, the correspondence between the motion of the tool in the frame of the camera and the motion of the robot in his base frame should be known. Moreover, if the programming by demonstration should be used for industrial robots, the interaction between the human and the robot must be limited. Consequently, the cell of the robot and the recording cell should be separated as shown in Fig. 3. A fixed marker is used to determine the working frame of the recording cell, and the camera can be move at a suitable position to record the motion. Making a correspondence between the fixed marker and the robot frame leads to the desired path for the robot.

III. EXPERIMENTS

A. Experimental setup description

In order to evaluate the system, an experimental setup is created at the Laboratory of Human Motion Analysis of the University of Liege[1]. Four 3D optoelectronic systems, *CX1* CODAmotion, based on active markers are used and considered as a gold standard reference system. In the following, the position of an object measured with these systems is considered as the ground truth. The markers are fixed on 3 elements: the camera, the fixed marker and the tool. It provides the position of the fixed marker and of the tool in the camera frame from which an estimation of the precision and the repeatability could be computed. The experimental setup is illustrated in Fig. 4

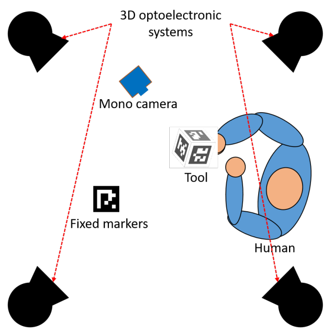


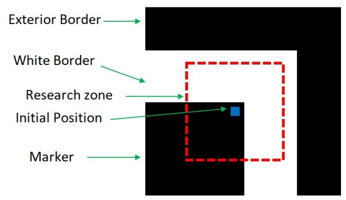
Fig. 4: Experimental setup developed at the Laboratory of Human Motion Analysis of the University of Liege and used to estimate the precision and the repeatability

B. Error sources

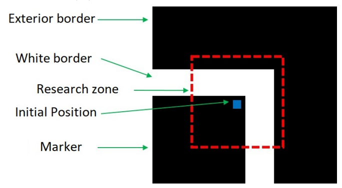
There are two main types of errors. They could be due to the distortion or to the numerical treatment of the images. The distortion means that the image is not a perfect perspective projection due to the usage of lenses and a calibration procedure is required which involves camera internal parameters[13]. However, some errors could be remain and the calibration may not be perfect. The second type of errors are due to the numerical resolution of the PnP problem and the determination of the corner positions.

[1]http://labos.ulg.ac.be/lamh/

1) The subpixel corner detection: The corner detection is improved using a subpixel method which should give a better estimation of the corner positions. The image is a discretization of the scene, so the initial guest for a corner position corresponds to a pixel. However, the real position of the corner is more precise using a subpixel method because the position is estimated using the pixels around the original guest to refine it. Nevertheless, the subpixel position is still an estimation which is degraded if the marker size decreases while keeping it at the same distance and if the distance increases while the marker size stays the same. Moreover, the marker should be surrounded by a white border to simplify its detection but if the border is to small it could induced some problems during the subpixel detection. The corner detected could be the external corner and not the corner of the marker. This situation is ambiguous, as it can be seen in Fig. 5. It can lead to important errors in some particular cases.



(a) Correct situation



(b) Ambiguous situation

Fig. 5: Two situations that could appear using the subpixel method

Generally several markers are detected in the images, so the errors of corner detection are compensated. However, if only one marker is detected, it can lead to strong orientation error. Even if the PnP problem has theoretically a unique solution, two orientations give similar images when they are projected. So a small error of corner detection can lead to the wrong orientation as it can be seen in Fig. 6.

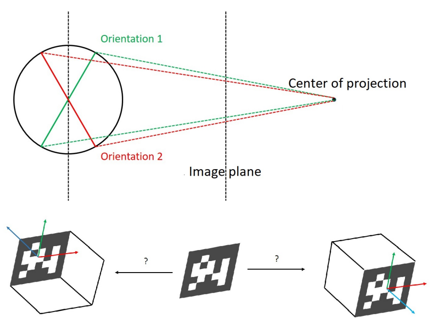


Fig. 6: The Z-flipping problem representation

In conclusion, to avoid the error due to the corner detection, several improvements can be used: the markers should be as big as possible with a large white border, the markers should be close to the camera, several markers should be detectable on an image or the image resolution should be increased. Also, the camera calibration should be done carefully with a camera that has small distortion.

C. Static measurements and quantification

Some static measurements have been done with the experimental setup. Twenty-five configurations have been measured with orientation and position variations and 100 images have been taken for each configuration. It allows evaluating the precision and the repeatability. The position error is computed using the formula $\| T - \hat{T} \|$ where T is the position vector obtained with the Codamotion system and $\hat{T}$ is obtained with the fiducial markers. For the rotational error, this formula can be used $\theta = acos\big((Trace(R^T\hat{R}) - 1)/2\big)$ where R is the rotational vector obtained using the Codamotion system and $\hat{R}$ is obtained with the fiducial markers.

The repeatability of the measurements is smaller than one millimetre if there is no error of corner detection. When some errors appear, it is in the range of 2-3 millimetres. However the precision is not as good as expected since it is in the range of the centimetre and around two degrees of errors. However, it seems clear that there is some calibration error since the precision of the tool and the fixed marker in the camera frame are worse than the precision of the tool in the frame of the fixed marker. In the three cases, the errors are in the range of the centimeter.

A more accurate procedure should be defined to study this phenomenon in static and in dynamic configurations. It will be done in a future paper.

IV. CONCLUSION

The programming by demonstration aims at bringing the practical knowledge of workers to the robot without the long development required by classical programming method. Consequently, it could be used to program robots used with small batch size, since the program could be changed easily. It requires a motion recording phase which can be done using an optical system. However, the optoelectronic systems are generally costly since they require several cameras and a synchronization system. The methodology presented in this article is based on a single 2D camera with square markers which is a cost-effective solution.

The first experimental measurements are promising. However, to validate the methodology, a study with dynamic motion should be done. Afterwards, a practical case should be realized using a robot to reproduce a recorded trajectory. It could require an adaptation since some motion done by the human could not be reproducible by the robot.

ACKNOWLEDGEMENT

The first author would like to acknowledge the Belgian Fund for Research training in Industry and Agriculture for its financial support (FRIA grant).

References

[1] A. Billard, S. Calinon, and S. Schaal, “Springer handbook of robotics,” Choice Rev. Online, vol. 46, no. 06, pp. 46-3272-46–3272, 2013.

[2] S. Garrido-Jurado, R. Muñoz-Salinas, F. J. Madrid-Cuevas, and M. J. Marín-Jiménez, “Automatic generation and detection of highly reliable fiducial markers under occlusion,” Pattern Recognit., vol. 47, no. 6, pp. 2280–2292, 2014.

[3] J. Wang and E. Olson, “AprilTag 2 : Efficient and robust fiducial detection,” pp. 2–7.

[4] R. Pellois, L. Joris, and O. Br, “Robot control based on human motion analysis with IMU measurements.”

[5] “Robot Learning from Demonstration in Robotic Assembly: A Survey,” Robotics, vol. 7, no. 2, p. 17, 2018.

[6] J. Ijspeert, J. Nakanishi, and S. Schaal, “Learning Attractor Landscapes for Learning Motor Primitives,” Adv. Neural Inf. Process. Syst. 15, pp. 1547–1554, 2002.

[7] V. Villani, F. Pini, F. Leali, and C. Secchi, “Survey on human–robot collaboration in industrial settings: Safety, intuitive interfaces and applications,” Mechatronics, vol. 55, no. March, pp. 248–266, 2018.

[8] S. Garrido-Jurado, R. Muñoz-Salinas, F. J. Madrid-Cuevas, and M. J. Marín-Jiménez, “Automatic generation and detection of highly reliable fiducial markers under occlusion,” Pattern Recognit., vol. 47, no. 6, pp. 2280–2292, 2014.

[9] F. Röhrbein, G. Veiga, and N. Ciro, “Gearing Up and Accelerating Cross-Fertilization between Academic and Industrial Robotics Research in Europe: Technology Transfer Experiments from the ECHORD Project ABC,” Springer Tracts Adv. Robot., vol. 94, no. January, 2014.

[10] P. Pastor, H. Hoffmann, T. Asfour and S. Schaal, ”Learning and generalization of motor skills by learning from demonstration,” 2009 IEEE International Conference on Robotics and Automation, Kobe, 2009, pp. 763-768. doi: 10.1109/ROBOT.2009.5152385

[11] Bradski, G.,” The OpenCV Library,” Dr. Dobb’s Journal of Software Tools, 200

[12] Y. Zheng, Y. Kuang, S. Sugimoto, and K. Åstr, “Revisiting the P n P Problem : A Fast , General and Optimal Solution,” 2013.

[13] Z. Zhang. ”A Flexible New Technique for Camera Calibration.” IEEE Transactions on Pattern Analysis and Machine Intelligence, 22(11):1330-1334, 2000.