



HAL
open science

Teachless teach-repeat: Toward Vision-Based Programming of Industrial Robots

Mathias Perrollaz, Sami Khorbotly, Amber Cool, John-David Yoder, Eric Baumgartner

► **To cite this version:**

Mathias Perrollaz, Sami Khorbotly, Amber Cool, John-David Yoder, Eric Baumgartner. Teachless teach-repeat: Toward Vision-Based Programming of Industrial Robots. IEEE International Conference on Robotics and Automation, May 2012, St Paul, Minnesota, United States. hal-00671209

HAL Id: hal-00671209

<https://inria.hal.science/hal-00671209>

Submitted on 16 May 2012

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Teachless teach-repeat: Toward Vision-based Programming of Industrial Robots

Mathias Perrollaz, Sami Khorbotly, Amber Cool, John-David Yoder and Eric Baumgartner

Abstract—Modern programming of industrial robots is often based on the *teach-repeat* paradigm: a human operator places the robot in many key positions, for teaching its task. Then the robot can repeat a path defined by these key positions. This paper proposes a vision-based approach for the automation of the teach stage. The approach relies on a constant auto-calibration of the system. Therefore, the only requirement is a precise geometrical description of the part to process. The realism of the approach is demonstrated through the emulation of a glue application process with an industrial robot. Results in terms of precision are very promising.

I. INTRODUCTION

Robots are extensively used in production lines, because they allow fast and precise processing. However, every time a production line is set to a new task, significant human effort is required in order to program each robot for its new process. Among possible programming paradigms, the teach-repeat approach is widely used: human operators have to position the robot in all the required configurations, in order to build their programmed path. During the later execution of the program, the robot repeats the learned path and will require all the objects to be placed very precisely in the same position. Considering the significant progress in robot perception and particularly in computer vision, this teaching stage could be partially automated, which would simplify the establishment of the production line and gain flexibility for the operation of the line.

In this paper, we propose a vision-based approach for automatically teaching and repeating the task of an industrial robot. Our approach does not require a precise initial calibration of all the components, because it performs re-calibration of the elements during the teach stage. During the repeat stage, the vision sensor is only used to check if some conditions have changed. In this case, a new teaching process is automatically performed. The only requirement is a precise design of the part to process, which is generally available as a CAD file.

Section II presents some related work. Our methodology is described in section III. An implementation is proposed in section IV, with the emulation of a glue application process. Section V provides some early experimental results. Section VI concludes and discusses future work.

M. Perrollaz (mathias.perrollaz@inrialpes.fr) is with the INRIA Grenoble.

J-D. Yoder (j-yoder@onu.edu), Eric Baumgartner, Amber Cool and Sami Khorbotly are with Ohio Northern University.

II. RELATED WORK

Over the last few decades, there have been great improvements in industrial robot programming. Early methods needed low level programming of the joints move. Now the programming can be performed at the tool, process or product level [1]. To work at the product level, some robot manufacturer propose software for 3D modeling of the workcell of a robot. Then the model is used to generate code for the robot to perform its task.

At the tool level, a very popular approach is the teach-repeat approach, sometimes referred as *leadthrough*. An operator jogs the tool to various key-points, using the teach-pendant of the robot, and can save the set of points to generate the processing program. Another teach-repeat methodology, known as *manual leadthrough* or *lead-by-the-nose*, lets the user move the robot tool by hand in order to record the processing path. With these two approaches, the repeat stage consists of making the robot repeat the learned path at higher speeds. A limitation is that if anything changes in the configuration of the environment, for instance if the object to process has moved, then the learned path is no longer correct. For this reason, the authors of [2] propose to use a vision system for learning some appearance information during the teach stage. This provides flexibility of the repeat stage, by allowing errors in positioning.

In all these methods, the human operator has a central role for the teach stage. Considering the recent progress in 3D vision, an attractive idea is to use a vision system for partial automatization of this teach stage. Visual servoing for manipulation has been widely studied [3], particularly for robot's interaction in human environment, for instance in grasping objects [4] But this problem has been somehow less studied in industrial robotics, where the visual sensors are more often dedicated to the control of process execution or for part inspection.

Theoretically, a finely calibrated stereo-vision system for detecting a part to process, combined with a very fine eye-hand calibration and a perfect knowledge of the part's geometry could replace the teaching stage. The precise part geometry is generally available as a CAD file for industrial parts. Eye-hand calibration for manipulation has been widely studied [5], as well as the calibration of a multiple cameras system [6][7]. Nevertheless, assuming the ideal conditions necessary for such a fully-calibrated

approach could be hazardous for industrial applications. Considering all the vibration, temperature variation, shock and other disturbances existing in a production line, it would be extremely difficult to maintain an optimal calibration of all the components. Therefore, a continual auto-calibration of the system is required.

Such approaches have been investigated for instrument placement, especially for the Mars Exploration Rover (MER) project, where no human intervention can be assumed. Several approaches have been proposed, derived from the method known as Camera Space Manipulation (CSM) [8]. The HIPS approach [9] proposes to refine the model parameters during the approach of the instrument, in order to ensure having a very precise local model around the target. AGATE [10] is a similar technique, with target selection being accomplished in camera-space. In the field of industrial robots, the authors of [11] propose to use CSM for path tracking, allowing to deal with deformed surfaces.

III. VISION-BASED TEACH-REPEAT

A. The approach

The idea behind our approach is to use a vision system to determine accurate positions of the tool in the robot's base frame, corresponding to a set of targets on the object to process. The repeat stage is completed by moving the robot along these positions. If the object to process is always placed in the same position, then this process can be repeated constantly. Otherwise, the new position of the object is estimated and the teaching stage is run again. The reason for running the teaching process again is that the calibration is locally very accurate, but globally less accurate. So that if the part moves substantially, only applying a translation and a rotation to the targets points would lead to a poor solution.

1) *Camera model*: In order to ensure sufficient accuracy in the estimation of the positions of the tool, our approach relies on a camera-space-based method, which constantly updates the estimated camera model parameters, similar to HIPS and AGATE. The main advantage of this approach is to provide a measurement model which can be very precise locally around the robots tool, rather than being precise globally. Therefore, it allows for very precise positioning. This method also offers constant auto-calibration of the system, without any human intervention.

The approach is based on the eighteen parameters CAHVOR vectorial camera model, which is similar to a pinhole with radial distortions, and includes extrinsic parameters [12]. The projection $\mathbf{u} = (u, v)$ in camera-space of a 3D point \mathbf{x} is described as:

$$\mathbf{u} = \begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} f_u(\mathbf{x}, \Gamma) \\ f_v(\mathbf{x}, \Gamma) \end{bmatrix} \quad (1)$$

with Γ representing the parameters of the camera model. With the CAHVOR model, $\Gamma = \{\mathbf{c}, \mathbf{a}, \mathbf{h}, \mathbf{v}, \mathbf{o}, \mathbf{r}\}$ and f_u and f_v are defined as as :

$$\begin{cases} f_u(\mathbf{x}, \Gamma) = \frac{(\mathbf{x}' - \mathbf{c}) \times \mathbf{h}}{(\mathbf{x}' - \mathbf{c}) \times \mathbf{a}} \\ f_v(\mathbf{x}, \Gamma) = \frac{(\mathbf{x}' - \mathbf{c}) \times \mathbf{v}}{(\mathbf{x}' - \mathbf{c}) \times \mathbf{a}} \end{cases} \quad (2)$$

with:

$$\begin{cases} \mathbf{x}' = \mathbf{x} + \mu \lambda \\ \mu = r_0 + r_1 \tau + r_2 \tau^2 \\ \tau = \lambda \lambda / \zeta^2 \\ \lambda = \mathbf{x} - \mathbf{c} - \zeta \mathbf{c} \\ \zeta = (\mathbf{x} - \mathbf{c}) \times \mathbf{o} \end{cases} \quad (3)$$

and $\mathbf{r} = [r_0, r_1, r_2]^T$.

The parameters Γ_i of camera i can be estimated from a set of points, whose coordinates are known in both camera space and physical space, through the minimization of:

$$\kappa_i = \sum_k [(u_k - f_u(\mathbf{x}_k, \Gamma_i))^2 + (v_k - f_v(\mathbf{x}_k, \Gamma_i))^2] W_k \quad (4)$$

The minimization procedure of equation 4 is detailed in [12].

In the spirit of the HIPS/AGATE methodology, the initial $\{\mathbf{c}, \mathbf{a}, \mathbf{h}, \mathbf{v}, \mathbf{o}, \mathbf{r}\}$ parameters are obtained by minimizing the quantity defined in equation 4 on preplan data. Then, during the motion of the robot's tool touching a target, additional measurements are collected. These measurements are added to the existing set of data and the camera model parameters are re-estimated. This strategy helps keep a precise local measurement model around the robot tool. The weighting factor W_k can be used to give particular confidence on certain samples: attributing a strong weight to points around a particular region of the camera space (e.g. around a target) can lead to a more precise model in this region, rather than globally.

2) *Coordinates systems*: Several coordinate systems are used in this work:

- Robot's coordinate system, or base frame (BF). Coordinates are noted \mathbf{x}^R in BF .
- Camera space (CS_i) associated to the camera i . Coordinates are noted \mathbf{u}^i in CS_i .
- Object frame (PF) are associated to the object to process or to its fixture (if any). Coordinates are noted \mathbf{x}^P in PF .

Relationship between BF and CS_i are given by equations 1, 2 and 3. The relationship between PF and BF is defined by an homogeneous matrix \mathbf{H}_P , which is determined based on the robot's joint angles and the robot's forward kinematics.

Our method uses multiple cameras in order to reconstruct 3D information from the measurements in camera-space. This reconstruction is obtained by minimization of the error of projection in camera-space. Let \mathcal{C} be the set of cameras

used in the system. The \mathbf{x} position of a point in 3D space is estimated through the minimization of:

$$\kappa_U = \sum_i [(u_i - f_u(\mathbf{x}, \Gamma_i))^2 + (v_i - f_v(\mathbf{x}, \Gamma_i))^2] \quad (5)$$

3) *Targets*: A set of targets T_i is defined in the object frame. These targets are extracted from the object geometry and from the process to apply to the object. A target is defined as a position \mathbf{x}_i^P in PF and an orientation of tool represented by a quaternion \mathbf{q}_i^P .

For each target T_i the teaching process intends to compute a robot tool pose, as a position \mathbf{x}_i^R in base frame and a quaternion for orientation \mathbf{q}_i^R .

4) *Description of the process*: The complete process can be decomposed as follow:

- 1) **initialization**: estimate the parameters of the set of cameras Γ_C ,
- 2) **detection**: detect the object's pose P_0 in camera space,
- 3) **teach**: for each target T_i on the object, estimate the tool pose $(\mathbf{x}_i^R, \mathbf{q}_i^R)$ in BF , while constantly updating Γ_C ,
- 4) **repeat**: if the current pose P of the object is close to P_0 , repeat the learned path, otherwise go back to detection step.

B. Initialization

The first stage of the method, referred as preplan, is the initial calibration for every camera involved. This is performed by positioning the robot tool in a large number of poses, and recording its position both in base frame and camera-space. To properly solve the minimization problem of equation 4, it is important to have a good initial guess. For this purpose, the extrinsic parameters of the camera are estimated by minimizing the projection error over the preplan dataset, while neglecting the distortion parameters. Thus the extrinsic matrix $\mathbf{H} = [\mathbf{R}|\mathbf{t}]$ is found through the minimization of:

$$\kappa_R = \sum_k [\mathbf{u}_k - \mathbf{K}\mathbf{H}\mathbf{x}'_k]^T [\mathbf{u}_k - \mathbf{K}\mathbf{H}\mathbf{x}'_k] \quad (6)$$

\mathbf{x}'_k being the homogeneous coordinates of \mathbf{x}_k . For that, it is necessary to have a first estimation of the camera intrinsic matrix \mathbf{K} , defined as:

$$\mathbf{K} = \begin{bmatrix} \alpha_u & 0 & u_0 \\ 0 & \alpha_v & v_0 \\ 0 & 0 & 1 \end{bmatrix} \quad (7)$$

with (α_u, α_v) representing the focal length expressed in pixels horizontally and vertically, and (u_0, v_0) being the position of the optical center on the image plane. \mathbf{K} can be obtained using a standard camera calibration method with a checker board. The calibrated extrinsic and intrinsic matrix are converted into a CAHV camera model, which is used as the initial guess for the estimation of the CAHVOR model parameters. The $\{\mathbf{c}, \mathbf{a}, \mathbf{h}, \mathbf{v}\}$ vectors are expressed in BF and defined as follow:

- The \mathbf{c} vector is the vector that points from the origin of BF toward the focal point of the camera : $\mathbf{c} = \mathbf{t}$
- The \mathbf{a} vector is the unit vector that points out from the camera's image sensor out through the lens of the camera.
- $\mathbf{h} = \alpha_u \times \mathbf{h}_0 + u_0 \times \mathbf{a}$, with \mathbf{h}_0 being the unit vector pointing along the horizontal direction of the image sensor.
- $\mathbf{v} = \alpha_v \times \mathbf{v}_0 + v_0 \times \mathbf{a}$, with \mathbf{v}_0 being the unit vector pointing along the vertical direction of the image sensor.

This set of parameters as well as the preplan data are fed into the CAHVOR estimation algorithm. Note that at this stage, the points are equally weighted for the minimization of equation 4.

C. Detection

The object to process is observed in CS_i , as a set of images points \mathcal{P}_i^U . It could be a limited number of visual cues on the object or its fixture, but it could as well be a large point cloud, specific patterns or image features. This only depends on the implementation choices.

D. Teach

The teaching stage consists of computing a precise positioning of the robot's tool with respect to each of the targets. Targets are initially defined in the object coordinate system, so their position must be computed in the base frame of the robot. In order to achieve maximum accuracy for this computation, the position is re-computed after all displacement of the robot while recursively refining the model of the cameras. Therefore, as the robot tool approaches the target, the model becomes more precise within vicinity of the target. This behavior can be reinforced by increasing the value of the weighting coefficient W_k defined in equation 4.

The approach trajectory is obtained by having the robot move toward the target iteratively by a proportion α of the remaining distance. The target is considered reached if the final distance is within the expected precision. Algorithm1 below details the teach stage.

E. Repeat

At the end of the learning process, the program has saved a set of tool positions and orientations in the base frame, thus defining the tool path for processing the object. Then, if a new object is placed in the fixture at the same position, the robot only has to move the tool along the path for processing the object. If after detection of the fixture, it appears that it has moved in camera space (i.e. either the cameras or the fixture have moved from their original position), the teaching stage is performed once again, to compute the new path. This ensures that the system is always correctly calibrated for the task.

Algorithm 1 Algorithm for learning joint-space positions for processing the object. (** represents the quaternion product.)

Require: initial Γ_C

Require: \mathcal{P}_C^U

$\mathbf{H}_p \leftarrow \text{estimate3DPose}(\mathcal{P}_C^U, \Gamma_C)$

for target (T_i) on the object **do**

$\mathbf{x}_t^R \leftarrow \mathbf{H}_p \times \mathbf{x}_i^P$

$\mathbf{x}_r^R \leftarrow$ get current position from the robot

$d \leftarrow \text{norm}(\mathbf{x}_r^R - \mathbf{x}_t^R)$

while $d > \epsilon$ **do**

$\mathbf{x}_m^R \leftarrow \alpha \times (\mathbf{x}_r^R - \mathbf{x}_t^R)$

$\mathbf{q}_p^R \leftarrow$ mat to quaternion(\mathbf{H}_p)

$\mathbf{q}_r^R \leftarrow \mathbf{q}_i^P * * \mathbf{q}_p^R$

move robot to $[\mathbf{x}_m^R, \mathbf{q}_r^R]$

$\mathbf{x}_r^R \leftarrow$ get current position from the robot

$\mathbf{u}^j, j \in \mathcal{C} \leftarrow$ observe robot position

add to data ($\mathbf{x}_r^R, \mathbf{u}^j, j \in \mathcal{C}$)

$\Gamma_C \leftarrow$ re-estimate CAHVOR models

$\mathbf{H}_p \leftarrow \text{estimate3DPose}(\mathcal{P}_C^U, \Gamma_C)$

$\mathbf{x}_t^R \leftarrow \mathbf{H}_p \times \mathbf{x}_i^P$

$d \leftarrow \text{norm}(\mathbf{x}_r^R - \mathbf{x}_t^R)$

end while

$\mathbf{x}_i^R \leftarrow \mathbf{x}_r^R$

$\mathbf{q}_i^R \leftarrow \mathbf{q}_r^R$

end for

IV. IMPLEMENTATION

We developed an implementation of our approach using a 6-axis industrial robot (fig. 1). The system uses a stereo pair of cameras for perception. The position of the robot tool in stereo camera-space is obtained via the detection of blinking light-emitting diodes (LEDs) placed on the robot's end-effector. Since this work does not focus in the detection of the object to process, the object is placed in a fixture equipped with visual cues.

A. Emulated task

The implementation emulates the robot arm applying glue on a metallic part, provided by our industrial partner, American Trim¹. The CAD geometry and technical designs were available, so that target points can be extracted from the design of the object itself. Note that the points are designated so that, in a first approximation, the trajectory between two successive points is linear. Then the method can be independant from the manipulator's model (every manipulator can perform linear moves).

B. Detection of the hand

The robot's end-effector is equipped with a LED blinking system. The blinking frequency is roughly a third of the frequency of the cameras to avoid possible aliasing. Unlike [13], we choose a rather simple design of the LED lamp

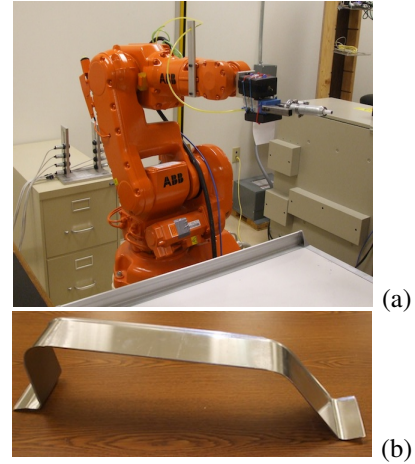


Fig. 1. The approach was implemented on a 6-axis industrial robot (a). A LED box is placed on the end-effector and a marker is used to emulate the robot's tool. The emulated process is the application of glue on a metallic part (b).

which does not require synchronization between the emitter (LED) and the receiver (camera). This approach is because the LED lamp is only used for the teach stage. Therefore the arm can stay in a static position during the acquisition of a set of images. Once the images are acquired, two images, respectively representing the mean value and the standard deviation over time of the intensity of each pixel, are computed, as illustrated on figure 2. The cyan component of both images are computed using the green and blue planes (because our LEDs produce cyan rather than blue light). Then connected components of high standard deviation are extracted. These regions are detected as LEDs if their mean value over the cyan image is also sufficiently high.

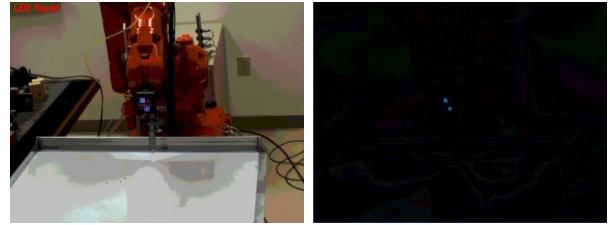


Fig. 2. Mean image (with reduced dynamic) and standard deviation image, used for detection of the leds.

C. Detection of the object

In order to detect the fixture of the object to process, and estimate its pose relative to the base frame, the fixture is equipped with 3 circular fiducials, as shown on figure 3. These visual cues have been used in several efforts. Note that we choose this kind of fiducials for ease and accuracy of detection. Other approaches could have been used, as the LED system presented above or a system of visual tags [14].

For detecting these cues, we decided to simultaneously detect regions of connected components of high intensity

¹www.amtrim.com

(white) and circular contours. Regions which respond to both detectors, and have a very close center of mass with both approach (less than 2 pixels error) are used as hypotheses. Then for each of these regions, a confidence is computed, based on its size and shape: a dip of intensity must be detected in 8 directions and centered on the center of the region. The 3 best regions according to this criterion are considered as the three cues of the fixture. The current camera models are used to retrieve their 3D position in base frame. Arbitrarily, we define the plane formed by the center of these cues as the plane of equation $Y_f = 0$ in the coordinate system of the fixture. The normal vector to this plane is obtained by cross product of two vectors formed by the 3 cues position. The axis formed by the two more-spaced cues defines the direction of the \vec{X}_f axis. Thus, the position and orientation of the fixture relative to the robot can be estimated for the current set of parameters of the cameras Γ_C .

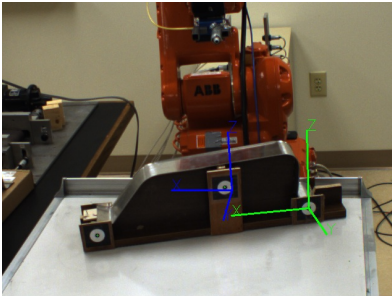


Fig. 3. Detected fixture of the object to process. The estimated coordinate system of the robot (after the initial estimation of CAHVOR parameters) is represented in blue, the coordinate system of the fixture is in green.

V. EXPERIMENTAL RESULTS

A. Hardware

The teachless teach-repeat approach has been tested with the 6-axis ABB IRB-140 robot shown in figure 1. The vision sensor is a pair of PointGrey Flea2 cameras, delivering 640×480 RGB images at 15FPS. Cameras are placed on a common portable support with a baseline of 15cm. While the cameras are almost parallel, no particular efforts were taken to ensure a proper rectified configuration, which is not required for our algorithm.

The communication between the robot and the PC is performed through the serial port, with a message-based protocol. The C++ program for perception, control and command runs on a desktop PC equipped with an INTEL XEON W3505 (2009), dual core 2.54GHz, with 4GB of RAM. The initialization is performed using a function from the OpenCV library, which is based on the Direct Linear Transform (DLT)[15].

We defined 14 target points out of the CAD file of the part, to test the process. The part was too large compared to the robot's work volume, so only half of the part was

processed. We choose a region with both flat and curved areas, to assess the robustness of our approach for these different shapes. The tool of the robot is simulated using a whiteboard marker, which roughly reproduces a glue dispenser tool. At the same time, it allows drawing the executed path on the object for measuring the accuracy of our approach.

It is important that the points chosen on the part are separated by a linear path, so that the repeat stage can be performed without problems.

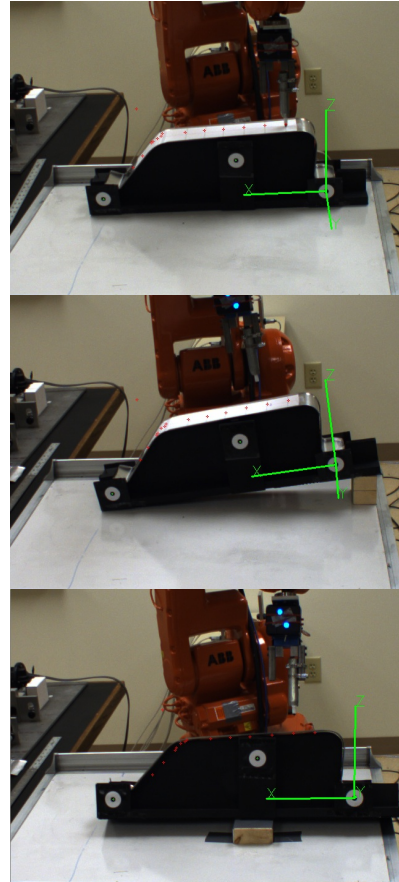


Fig. 4. Examples of target points during the teach stage.

B. Precision

For estimation of the precision of our system, we performed the teach-repeat with various positions of the fixture. Using a marker as tool, the path is drawn on the part during each repeat stage. We placed a sheet of paper on the fixture, with marks corresponding to target points, allowing to roughly measure the precision. Figure 5 shows the results for 6 different positions/orientations of the fixture. Note that, for the teaching stage, the tool moves from right to the left while approaching the target. This explains the portions of lines seen in figure 5. The actual target always correspond to the left-most point of each segment. Centimeter-level precision is reached, as the maximum distance between two realizations is 1cm. For some cases, no

drawing appears on the part because the tool was too high (of about 5mm). This is clearly not sufficient for all industrial purposes, but it is promising. Indeed, considering how the marker and LED lamp are mounted on the arm, the wrist to tool transformation changes over time. Since the concept was first demonstrated, a finer experimental evaluation will be set-up, with the objective to reach millimeter-level precision. The curvy part of the path also causes challenges, because in current implementation the robot can only follow straight line trajectories. It shows that an implementation of circle trajectories would be necessary for having smoother motions.

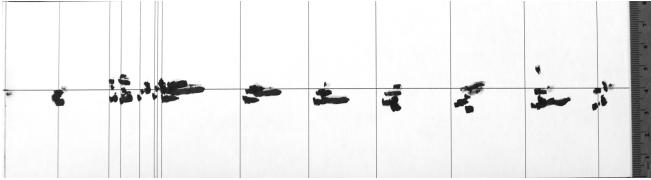


Fig. 5. Precision obtained for 6 positions of the fixture. The central line corresponds to the expected path. The vertical lines show expected longitudinal position for targets. For each mark, the final position of the tool is the left-most point of the mark. The ruler on the right is graduated in centimeters.

Figure 6 shows the repeated trajectories corresponding to the taught positions from figure 5. A bias is visible, which may confirm the relative error of positioning for the marker and LED lamp.

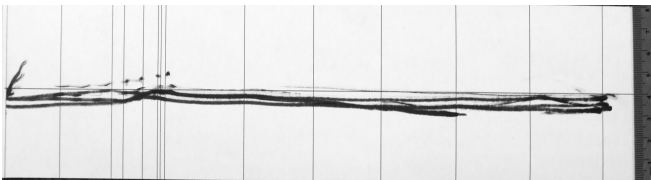


Fig. 6. Repeat trajectories obtained for the taught positions from figure 5.

C. Processing time

The complete time for the teaching stage was around 4 minutes 35 seconds. The majority of this time is used for acquisition of 10 images at each pose for LED detection, and for communication with the robot. As a comparison, the average computation time for the critical parts of the algorithm are:

- CAHVOR estimation : 90ms
- LED detection : 8ms
- 3D fixture localization : 45ms

While repeating the taught path, the robot is set to move at higher speed. Thus the repeat time is 6s. Note that all the target points are sent to the robot before the processing, so that the robot can achieve a smooth motion, without waiting for data.

VI. CONCLUSION AND FUTURE WORK

In this paper, we presented a new approach for programming the task of an industrial robot. Compared to

the classical teach-repeat methodology, our approach intends to replace the human programmer by a vision system during the teach stage. The strength of the proposed approach is that it relies on iterative re-estimation of the parameters of the system, hence avoiding reliance on a very precise initial calibration of the system. The validity of the approach has been demonstrated with an actual industrial robot emulating the application of glue on a metallic part. Early results concerning precision and time efficiency are very promising. After a more precise set-up of the tool and LEDs on the robot, we will initiate a large serie of experimentations with our prototype, in order to identify possible improvements.

There remains much work to do before considering an actual industrial use of *teachless teach-repeat*. Particularly, the automatic extraction of the target points from a CAD file, and the handling of circle trajectories will bring it closer to industrial standards. Additionnaly, replacing the fixture with a 3D points detection and registration algorithm would bring more flexibility. However, a significant step has been made toward a new way to program industrial robots.

ACKNOWLEDGMENT

The authors would like to thank Steve Couvillon and Larry Wilkerson of American Trim for supporting and motivating our work as well as for their advice.

REFERENCES

- [1] M. Hgele, L. Nilsson, and J. Pires, *Industrial Robotics*, in *Handbook of Robotics*, B.Siciliano and O. Khatib, Eds. Springer, 2008.
- [2] Y. Maeda and Y. Moriyama, "View-based teaching/playback for industrial manipulators," in *Proc. IEEE ICRA*, Shanghai, China, 2011.
- [3] D. Kragic and H. Christensen, "Survey on visual servoing for manipulation," *Computer*, vol. 15, 2002.
- [4] Y. Pang, Q. Huang, D. Jia, Y. Tian, J. Gao, and W. Zhang, "Object manipulation of a humanoid robot based on visual servoing," in *Proc. IEEE/RSJ IROS*, San Diego, USA, 2007.
- [5] Z. Zhao, "Hand-eye calibration using convex optimisation," in *Proc. IEEE ICRA*, Shanghai, China, 2011.
- [6] G. Harger, "A modular system for robust positioning using feedback from stereo vision," *IEEE Trans. on Robotics and Automation*, vol. 13(4), 1997.
- [7] N. Hollinghurst and R. Cipolla, "Uncalibrated stereo hand-eye coordination," *Image and Vision Computing*, vol. 12(3), p. 187192, 1994.
- [8] S. Skaar, W. Brockman, and W. Jang, "Three-dimensional camera space manipulation," *Int. J. Rob. Res.*, vol. 9, August 1990.
- [9] M. Robinson, E. Baumgartner, K. Nickels, and T. Litwin, "Hybrid image plane/stereo (hips) manipulation for robotic space applications," *Autonomous Robots*, vol. 23, 2007.
- [10] J.-D. Yoder and M. Seerlinger, "Long-range autonomous instrument placement," in *Proc. ISER*, Rio de Janeiro, Brazil, 2006.
- [11] E. Gonzalez-Galvan, C. Chavez, I. Bonilla, M. Mendoza, L. Raygoza, A. Loredo-Flores, and B. Zhang, "Precise industrial robot positioning and path-tracking over large surfaces using non-calibrated vision," in *Proc. IEEE ICRA*, Shanghai, China, 2011.
- [12] D. Gennery, "Least-square camera calibration including lens distortion and automatic edition of calibration points," in *Calibration and orientation of cameras in computer vision*. Berlin: Springer, 2001.
- [13] A. von Arnim, M. Perrollaz, A. Bertrand, and J. Ehrlich, "Vehicle identification using infrared vision and applications to cooperative perception," in *Proc. IEEE IV*, Istanbul, Turkey, 2007.
- [14] E. Olson, "Apriltag: A robust and flexible visual fiducial system," in *Proc. IEEE ICRA*, Shanghai, China, 2011.
- [15] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2003.