Int J Adv Manuf Technol (2016) 85:57–69
DOI 10.1007/s00170-014-6026-x

ORIGINAL ARTICLE

Stereo-based real-time 6-DoF work tool tracking for robot programing by demonstration

Marcos Ferreira · Paulo Costa · Luís Rocha · A. Paulo Moreira

Received: 2 October 2013 / Accepted: 5 April 2014 / Published online: 13 June 2014
© Springer-Verlag London 2014

Abstract This contribution presents a new system for fast and intuitive industrial robot reprogramming. It is based on a luminous marker built with high-intensity LEDs, which are captured by a set of industrial cameras. Using stereoscopy, the marker supplies 6-DoF human wrist tracking with both position and orientation data. This marker can be efficiently attached to any working tool which then provides a way to capture human skills without further intrusion in the tasks. The acquisition technique makes the tracking very robust against lighting conditions so no environment preparation is needed. The robot is automatically programmed from the demonstrated task which delivers complete abstraction of programming concepts. The system is able to perform in real time, and is low-cost starting with a single pair of industrial cameras though more can be used for improved effectiveness and accuracy. The real-time feature means that the robot is ready to perform as soon as the demonstration is over which carries no overhead of reprogramming times. Also, there is no interference with the task itself since the marker is attached to the work tool and the tracking is contactless; the human operator can then perform naturally. The test bed is a real industrial environment: a spray painting application. A prototype has been developed and installed, and is currently in operation. The tests show that the proposed system enables transferring to the machine the human ability of manipulating a spray gun.

Keywords Robotics · Programming-by-demonstration · Industrial manipulator · Stereoscopy · Tracking · Artificial vision

1 Introduction

Industrial manipulators are the ultimate automation tool. These machines deliver accuracy and repeatability, aiding industrial processes at becoming increasingly efficient while reducing production costs. As production lines tend to evolve into the concept of mass customization, i.e., working on small series with adapted and specialized procedures to each of them according to costumer specific needs, higher versatility is mandatory. Yet, programming industrial manipulators is still extremely time consuming and usually require experienced and highly qualified labor. The most common programming method involves the teach pendant: the programmer moves the robot to the desired position, specifies the velocity and type of movement, and repeats the process for the remaining way-points of the entire trajectory. Each point may even have to be tuned after an experiment run. Overall, this is not compatible with flexible setups neither with small companies' budgets since both qualified programmers and reconfigurations downtime imply strong financial efforts. Despite this programming complexity, manipulators are still strongly desired at production lines due to a series of advantages over human work, e.g., the ability to work continuously while preserving output quality, immunity to fatigue, distractions, and hazardous environments. As industrial processes evolve, so does the need for more advanced and flexible solutions. Also, skilled factory operators, whose expertise has been developed during many years of industrial practice, hold a strong know how on the processes. Such knowledge is hardly captured using software packages and graphical applications such as simulators and CAD (computer-aided design)

Electronic supplementary material The online version of this article (doi:10.1007/s00170-014-6026-x) contains supplementary material, which is available to authorized users.

M. Ferreira (✉) · P. Costa · L. Rocha · A. P. Moreira
Faculty of Engineering, Institute for Systems and Computer Engineering, University of Porto and INESC-TEC, R. Dr. Roberto Frias 378, 4200-465 Porto, Portugal
e-mail: marcos.a.ferreira@inescporto.pt

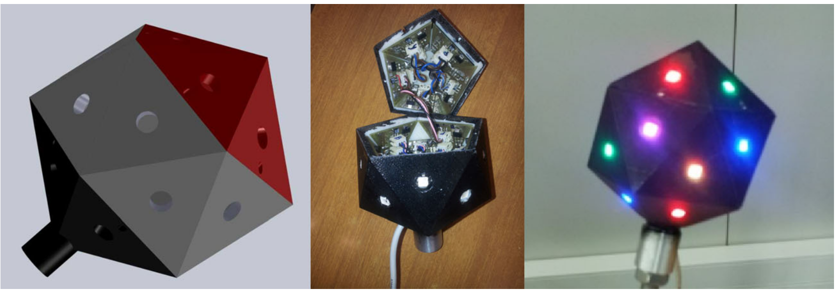


Fig. 1 CAD model of the designed marker (*left*), hardware details on the developed marker (*center*): cabling, electronics, and LEDs; the top hatch is held by screws to allow access to the interior (*right*) real marker with LEDs in ON state

based solutions. Solutions for acquiring the skill directly from the operators' tasks are of great interest.

Flupol is a Portuguese small/medium enterprise (SME) specialized in spray coating of diverse parts. Only two operators, with over 25 years' experience, are currently able to apply every coating since it requires fine technique. Robotization is mandatory to maintain competitiveness in the global market but the cost of reprogramming the robot for every part is simply unaffordable. A previous approach, presented in [1], makes use of both camera/laser triangulation to detect different parts and CAD-based programming to generate painting trajectories and adapt them to different dimensions. As highlighted before, the true skill is found in the operators' expertise developed over the years so it is desirable to grab, record, and playback that knowledge rather than trying to recreate it on the CAD interface.

1.1 Proposed solution and aims

In this line of thought, the work presented in this contribution proposes a methodology for fast industrial robot programming via human demonstration of work tool handling. The main goal is to achieve a new type of system that enables the human to quickly show the robot how to do a concrete task with abstraction of the programming language, and even completely avoiding the use of the teach pendant. The focus of this paper is to describe a new method for programming by demonstration (PbD) suitable for industrial operation. It makes use of a new luminous 6-DoF marker that is captured by a pair of industrial cameras. The artificial vision system captures images that can accurately tell us information about position and orientation. The resulting set of points that describe the human path are automatically transformed into a robot program. The human uses his natural abilities and skills to accomplish the demonstration process without needing further training on using new software packages, interfaces, or tools. Unlike similar researches, the proposed marker is able to perform in real time and the required apparatus is reduced and cheap.

1.2 Related work

The universe of human-robot interaction is vast and it is hard to cover all the current streams of research in the area. With a shorter scope, on motion and task demonstrations and industrial robot interface, there are still a number of contributions to consider. CAD-based programming is a major stream (and it is so with greater relevance in robotic painting applications); extensive examples are given in [2–5] (painting), and also in [6–8] (welding and generic CAD applications); in each of the previous, the user interacts with a simulated environment in order to draw the robot paths. These approaches present no actual method for the operator to demonstrate but rather a simplified and intuitive way to draw paths. Gesture recognition as presented in [9, 10] is another extensively covered subject. It serves well the purpose of interacting with the robot in an intuitive way since gestures are one of the most common/intuitive human communication tools. Yet, in such solutions, the robot must be preprogrammed and the gestures are used to deploy actions—the robot programmer cannot be

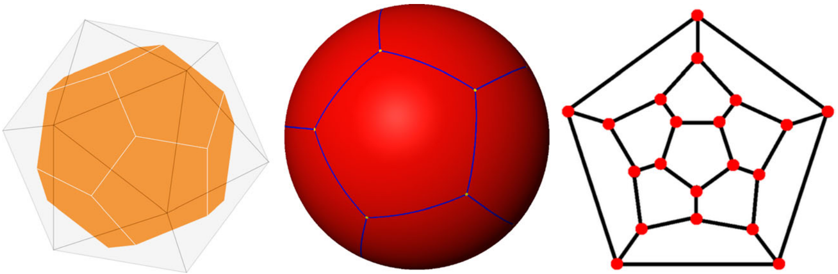


Fig. 2 The dodecahedron inscribed in an icosahedron (*left*). The marker outline follows the icosahedron shape while the LEDs are positioned on the vertices of a dodecahedron that touch the face centers of the former (*center*). The dodecahedron circumscribed sphere. All vertices lie on a spherical surface. A planar representation of the dodecahedron where no edges cross each other (*right*)—called a Schlegel diagram

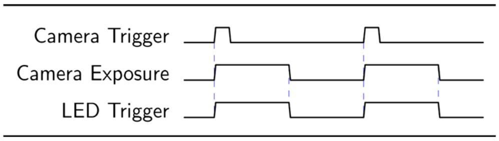


Fig. 3 A timing diagram depicting the synchronous image acquisition. Both cameras and the LEDs are triggered at the same time. Then, the cameras acquire the scene for a limited time (exposure time). The ultra-bight LEDs remain lit only during the camera exposure period

dismissed at the initial stage; the problem (and solutions) focuses on pattern recognition rather than actually controlling the robot movements. On more similar approaches to the work of these contributions, there is the vast work of R. Dillman, using stereo vision and a data glove [11] for tracking, and adding tactile sensors for grasp recognition [12–14]. Here, the movement of the human hand can be acquired but the apparatus, i.e., the glove lacks the industrial robustness, and it is not convenient to make an operator to wear such a device. Another similar approach has been presented by Hein and Worn [15, 16], where a new tracking marker is also proposed, based on infrared (IR) LEDs that can be coupled to any work-tool. The LEDs are captured by an array of infrared cameras. Our focus is on the development of a new marker that uses visible light LEDs; with a new synchronized acquisition, the visible markers can perform faster than the infrareds (either passive or active) because the former are easily distinguished using colour. The proposed solution is also cheap unlike most of the previous solutions as it is employed with a minimum of just two standard industrial cameras. Field et al. [17] contributed with a fairly recent survey (2011) on motion capture sensors for robotic applications. From this study, one can keep track of different technologies for body tracking: Elgammal and Lee [18], and Shon et al. [19] use passive IR markers for outdoor body tracking and indoor motion imitation by humanoid robots, respectively; Naksuk et al. [20] and Sigal et al. [21] use active IR markers—the former also directs the study towards human-like humanoid movements while the latter proposes algorithms to evaluate articulated human motion. Both passive and active IR markers have large post processing times. The computational complexity further increases with the number of individual markers. Marker-less options also exist [22, 23]; the authors propose a stereo-based human body tracking with no markers required though the precision of such solution falls short to the marker-based alternatives. Inertial [24] and magnetic [25] solutions (nonvision based) are also heavily explored. While magnetic sensors fail abruptly near metallic structures producing highly noisy measures, inertial sensors do require double integration and as such position estimation is poor. Moreover, data fusion is often required as an extra step [26]. In comparison, this paper contributes with a new method for hand motion tracking with a robust, real-time, and nonintrusive technique which is accomplished with standard and cheap industrial cameras.

2 Real-time 6-DoF work tool tracking

Successfully tracking the work tool pose during a given task is the primary step to achieve the desired human-robot skill transfer capability. We propose a simple but effective marker that delivers high-quality pose estimates. The device is based on a set of luminous markers (high-intensity LEDs) that enables video tracking using a minimum of two cameras. The shape of the marker and the choice of a singular pattern of colors for the LEDs are the key advantages to guarantee reliable and robust data from a human demonstration, which ultimately leads to a precise robot mimic.

The major concerns for the development of this new tracking tool were to achieve a device that provides an accurate measure of the pose of the human hand/tool while maintaining costs low, with reduced processing times and low impact on the process and on the human movements.

The following sections provide a thorough description on this tracking scheme: the hardware details of the proposed marker, the detailed software implementation of the real-time 6-DoF (degrees of freedom) pose measurements, and finally the robot code generation using the acquired data.

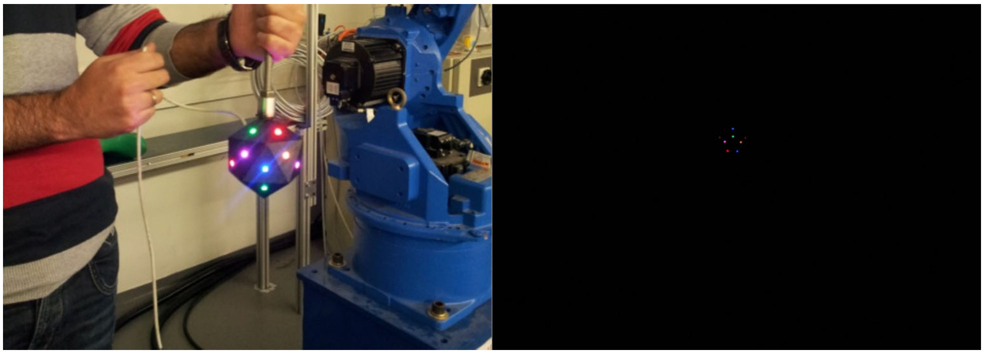


Fig. 4 Scene captured by a hand-held camera featuring the luminous marker (*left*). Synchronous capture of the previous scene (*right*)—the *Sincrovision* effect

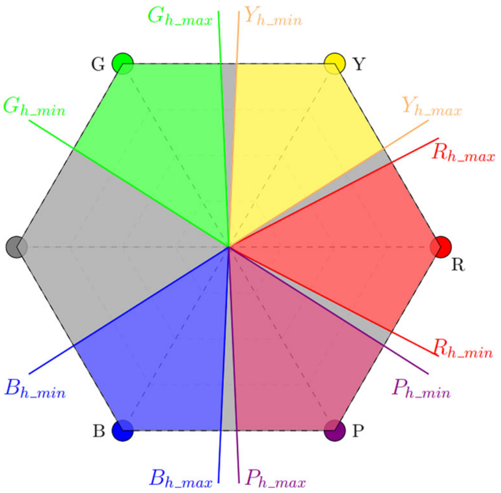


Fig. 5 Cluster classification using HSV color model. The LEDs are set to five different colors, with widely spaced hue values

2.1 Marker description—hardware

While one LED is enough to keep track of position (3D), to capture the other three degrees of freedom from orientation, at least three non-collinear LEDs are needed. Nevertheless, such a scarce number of lights would fail to provide a complete freedom of movements to the end-user: all of those individual markers should have to be visible at all times on both cameras; otherwise, pose estimation would fail due to occlusions. Increasing the number of cameras around the working area can fight back this problem though with a greater financial cost. On this line of reasoning, the proposed marker is based on 20 visible-light (RGB) LEDs. These are distributed in a special manner, based on the shape of an icosahedron (regular 20 face polyhedron), as it showed to provide an interesting set of properties that aid in constructive and algorithmic aspects:

- Figure 1 (left) shows the CAD model of the marker. The real model can be manufactured using a 3D printer. It is cheap and the overall weight is small. Different material densities can be used trading off weight for mechanical robustness.
- The icosahedron shape guarantees enough space to hold the LEDs and electronics. As depicted in Fig. 1 (center), the interior can house PCBs (printed circuit boards) with the required components to drive the RGB LEDs. The faces are ideal to attach the PCBs, forcing the LEDs to lie on the face centers.
- Placing the LEDs on the centre of each face covers full 360° rotations in every axis; there are always enough visible lights on both cameras so that it is possible to compute orientation and position. The number of cameras can then be kept to a minimum of two (for stereoscopy).
- The icosahedron is a regular (Platonic) polyhedron. This means that the LED distribution is symmetric. As such, position and orientation are two different matters, i.e., their computation is decoupled (the solid center is invariant under rotation).
- Connecting the face centers of an icosahedron results in the dual polyhedron, the dodecahedron (see Fig. 2 (left)). All vertices of a dodecahedron lie on a sphere (Fig. 2 (center)). Knowing the position of each vertex, one can find the center of the marker by means of estimating the best fitting sphere (which can be done computationally fast; more on this topic in the forthcoming Section 2.3.1).
- Another advantage of the icosahedron/dodecahedron shape is the representation of the spatial distribution of the LEDs in a planar diagram. Such representation is called a Schlegel diagram, Fig. 2 (right). It is useful for finding a color pattern in the LED distribution that simplifies the orientation estimation. From the Schlegel diagram, it is possible to distribute colors in such a

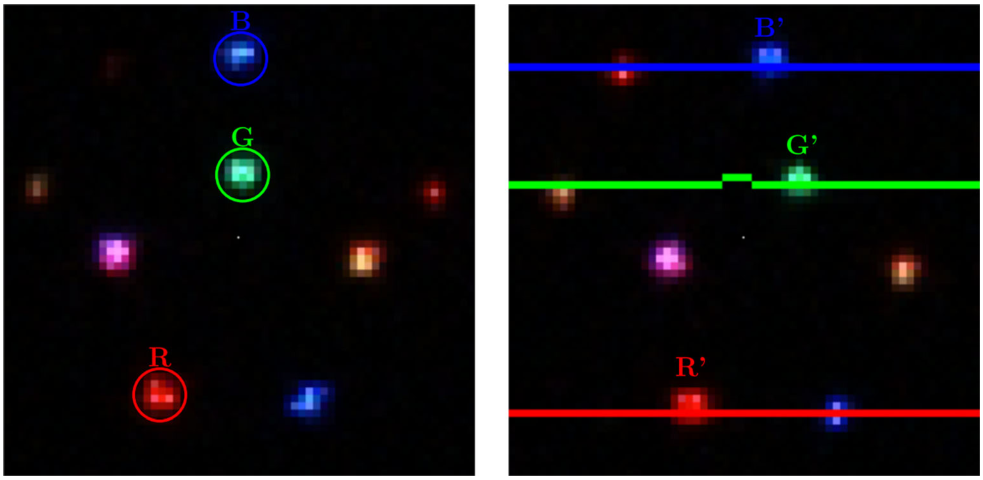


Fig. 6 Camera 1 zoomed image with a set of clusters (*left*); camera 2 zoomed image with epipolar lines (*right*); the *red*, *green*, and *blue lines* (not straight due to barrel distortion) are epipolar lines from the highlighted/labeled clusters in the image on the left

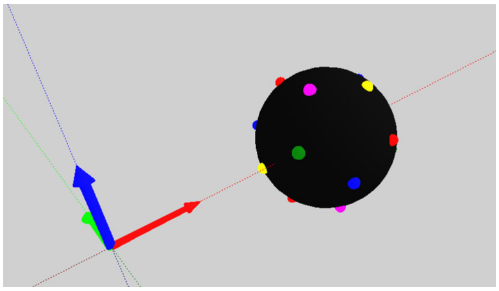


Fig. 7 OpenGL 3D scene with LEDs and a sphere. To estimate the whole marker position, each individual LEDs contributes for a sphere fitting algorithm

way that each region of the marker is different from every other from the cameras point of view (more on this topic ahead, in Section 2.3.2).

2.2 Marker detection

2.2.1 Stereoscopy using the "sincrovision" technique

The *sincrovision* concept was developed and patented by Malheiros et al. [27] at the University of Porto—Faculty of Engineering. It implements a system of 3D acquisition based on stereoscopic vision synchronized with high-intensity luminous markers. The key idea is to turn on the markers as soon as the cameras start acquiring image and turn them off after the camera exposure time has expired. Figure 3 shows a timing diagram of the system. The high-intensity lights will be very bright on the images while the background noisy data will have no time to be acquired by the camera. At the same time, blinking the markers for a short time makes it possible to stare at them; keeping them always on would cause eye damage. To ensure that unwanted light is not captured, the lenses aperture is reduced to a minimal.

This setup makes it possible to triangulate the individual markers' positions in space in a robust way, independently of lighting conditions in the scene thus ignoring most of the common noise sources in artificial vision applications. Figure 4 shows a typical image captured by the cameras using this synchronous feature: on the left, the scene is captured using standard camera aperture and exposition time; on the right, the *sincrovision* effect over the same scene. In terms of image processing, finding bright color clusters on a dark background is straightforward. A basic (and fast) global threshold for the pixels brightness can be robust even in the presence of intense artificial lighting such as ceiling lamps.

2.2.2 Solving the stereo correspondence problem

From a set of corresponding image frames, retrieving 3D data requires solving the stereo correspondence problem, i.e., knowing which pixel in image 2 corresponds to a given pixel in image 1. The first stage is the classification of each image cluster according to its color. As such, the LEDs are driven with different configurations (electronics) so that five distinct colors are achieved, as depicted in Fig. 5. The hue, saturation, and value (HSV) color space is useful for this classification because the hue value of each cluster can be easily and quickly computed. Also, the three primary colors and two secondary colors have widely spaced hue components which makes the color classes linearly separable in the hue space. The cyan was dropped since it was misclassified as blue or green in a large number of runs during the initial tests.

The second stage starts from a color-ordered cluster list from each image. The clusters are then matched according to their color and the geometric restriction provided by the stereo arrangement. Multiple view geometry, by Hartley and Zisserman [28], holds a complete guide for camera models, the stereoscopy principles, and camera calibration, which are the necessary basis for retrieving 3D measures from a pair of images. Simply put, a calibrated stereo setup holds the geometric description of the cameras' pose and their intrinsic parameters; given a pixel x from image 1, the corresponding pixels x' from image 2 lie on a line called epipolar. Matching the color clusters resolves into searching the epipolar lines for a cluster of

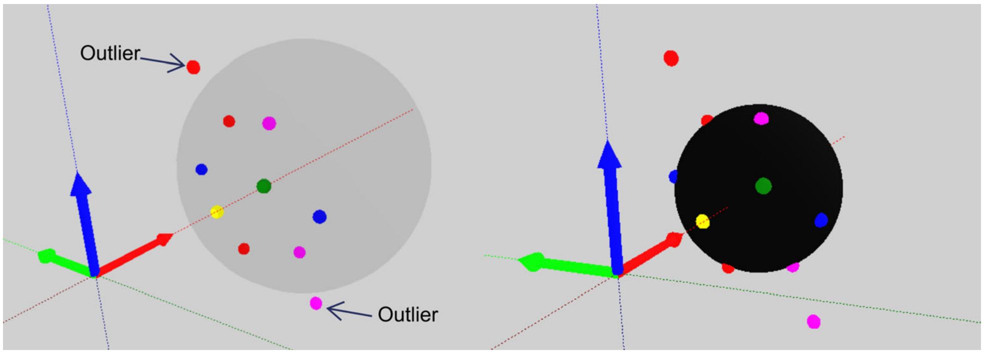


Fig. 8 Bad sphere fitting result (*left*). Outliers (due to bad stereo matches) make the radius deviate from the true known value (*right*). Iteratively dropping points and redoing the sphere fitting ultimately leads to the correct sphere estimation

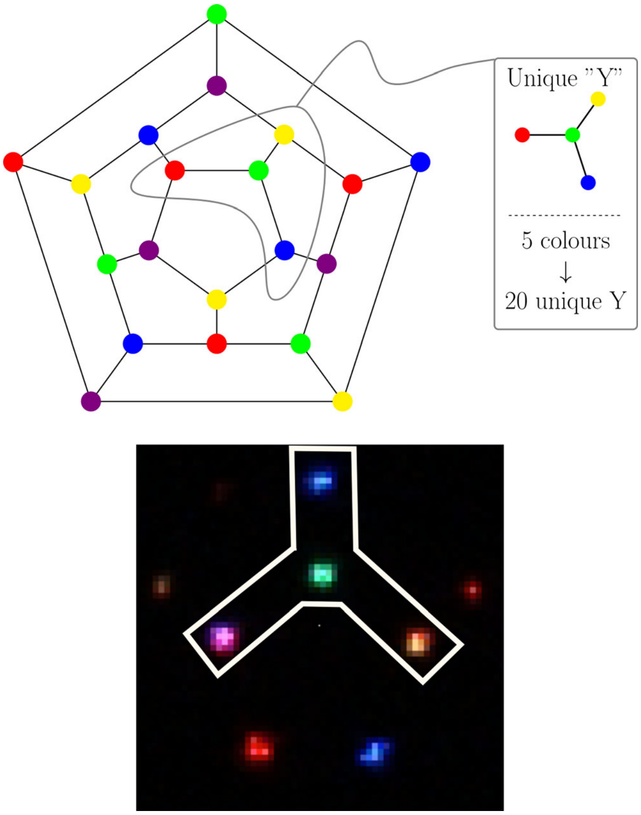


Fig. 9 The dodecahedron colored Schlegel diagram (*up*). It provides an intuitive way to view the distribution of LEDs around the dodecahedron. Using five colors allows a quick identification of the marker orientation as soon as a Y is completely visible in the images. Detection of a complete Y in a real image of the luminous marker (*bottom*)

the same color. Figure 6 shows a pair of images (zoomed) from the luminous marker where the matching process is highlighted. When this process is finished, a list of 3-D world coordinates of each LED position is obtained.

2.3 Marker (work tool) pose estimation

2.3.1 Position estimation

One of the advantages of the icosahedron-shaped marker lies on the position of the LEDs. Placing them on the center of each face enables that there are always some visible LEDs on both cameras for whichever movement is made by the human holding the marker (unless he steps in-between the marker and the cameras). Also, when the LEDs are positioned on the faces, they lie on a sphere shell that touches every face center. Due to this property, the 3-DoF related to the marker translation is computed taking advantage of this spherical positioning of the LEDs. Since the 3D coordinates of the matched clusters are already available (x_i,y_i,z_i), these world points are used to estimate the sphere shell at which they lie. From the sphere equation, with center at (x_c,y_c,z_c):

$$(x_i{-}x_c)^2+(y_i{-}y_c)^2+(z_i{-}z_c)^2=r^2 \qquad (1)$$

we use algebraic fitting [29] which solves the sphere fitting problem using least squares (the alternative method is the geometric fitting, which requires an iterative minimization [30]):

$$\kappa=x_c{}^2+y_c{}^2+z_c{}^2-r^2 \qquad (2)$$

Replacing (2) in (1) holds:

$$\underbrace{\begin{bmatrix} 1 & 2x_1 & 2y_1 & 2z_1 \\ \vdots & \vdots & \vdots & \vdots \\ 1 & 2x_n & 2y_n & 2z_n \end{bmatrix}}_{\mathbf{A}} \underbrace{\begin{bmatrix} -\kappa \\ x_c \\ y_c \\ z_c \end{bmatrix}}_{\theta} = \underbrace{\begin{bmatrix} x_1^2+y_1^2+z_1^2 \\ \vdots \\ x_n^2+y_n^2+z_n^2 \end{bmatrix}}_{\mathbf{b}} \qquad (3)$$

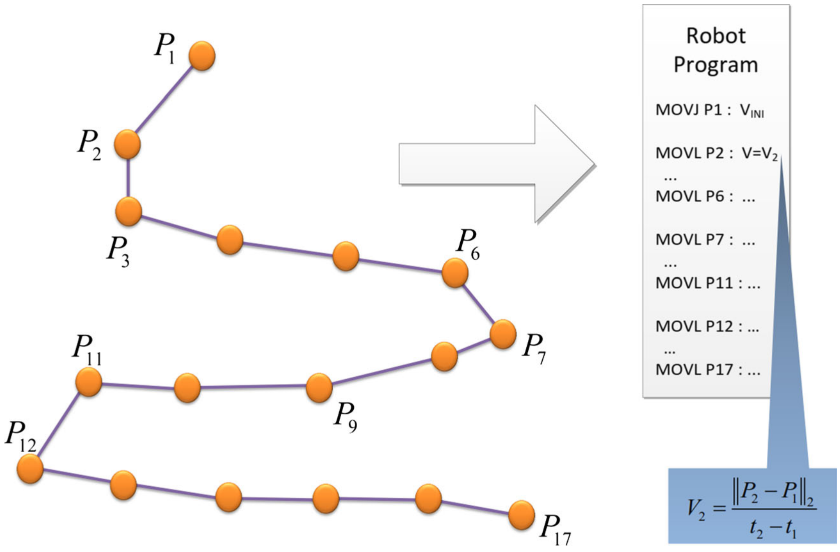
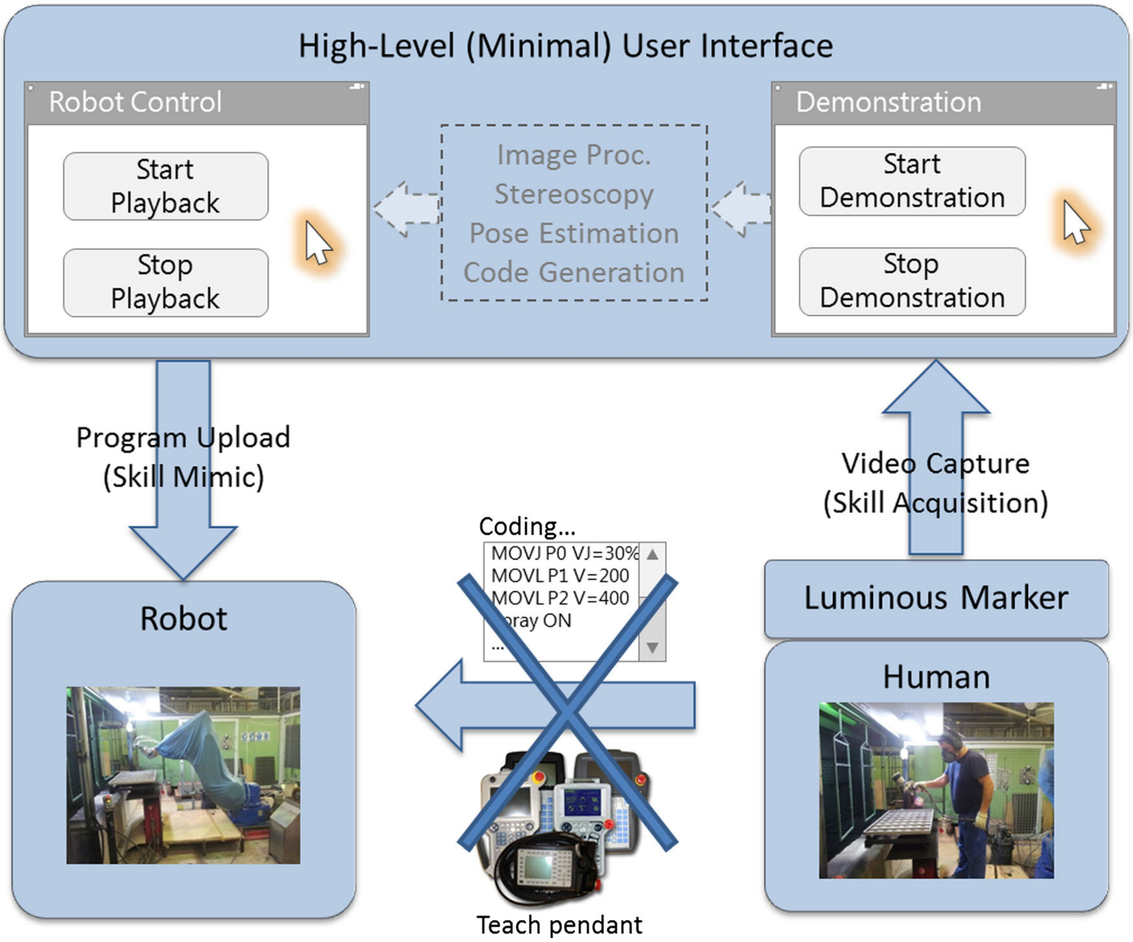


Fig. 10 The automatic generation of a robot program from the 6D tracking data. Each point is mapped into a move-linear instruction and the time stamp allows the definition of each segment speed



```
MOVJ P0 VJ=30%
MOVL P1 V=200
MOVL P2 V=400
pray ON
...
```

Fig. 11 The automatic generation of a robot program from the 6D tracking data. Each point is mapped into a move-linear instruction and the time stamp allows the definition of each segment speed

Figure 7 shows a 3D scene with the luminous markers and the sphere shell in which they lie at.

The advantage of the fast least squares solution (3) is that it can be used a number of times for the same image pairs to eliminate stereo ambiguities without compromising the real-time capability or minimization divergences: if some points are phantom, i.e., result from a bad stereo matching (outliers, marked in Fig. 8 (left)), using the algebraic fitting recursively while dropping some points ultimately comes up with the correct sphere estimation. Knowing when an estimate is good or not is done through the radius estimation: the marker radius is a well-known constructive measure and the bad stereo matches produce a radius estimate different from the known value (again, refer to Fig. 8 (left)); the outliers make the estimated radius too large. When the radius closes up to the real value, the estimate is considered a good measure and the marker position is known (Fig. 8 (right)); the real sphere is estimated with success after excluding some stereo measures recursively.

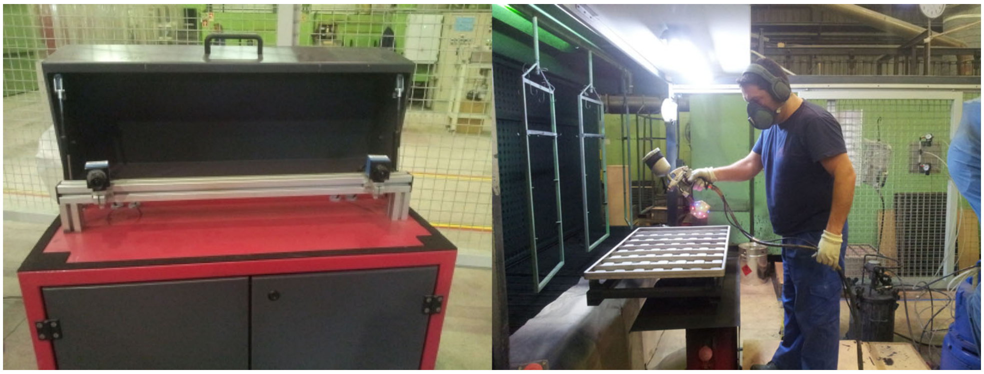


Fig. 12 The figure shows a cabinet that holds two cameras (*left*). These are placed 700 mm apart, and slightly rotated towards each other (*right*). Industrial cell for spray coating of baking parts

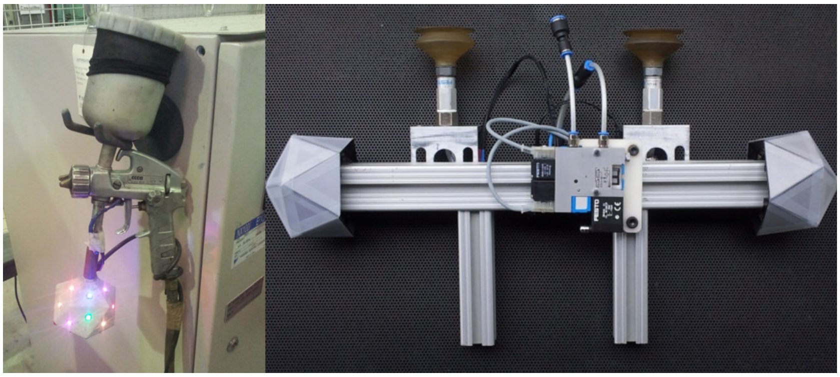


Fig. 13 Marker attached to industrial tools: spray painting gun (*left*); suction cups for bending operations (*right*)—one marker at each side of the tool, to minimize occlusions when holding large metal sheets

2.3.2 Orientation estimation

After retrieving the translation vector, the missing 3-DoF with respect to angular displacement is computed from the known well-matched 3D points.

Using again the advantages of the icosahedron, it is possible to build a list of the LEDs' coordinates in the 3D world, before and during the movement. When the icosahedron face centers are connected, we get the dual polyhedron, the dodecahedron. Each face center of the icosahedron is a vertex of the dodecahedron. There is an orientation of the marker at which the LEDs lie at the given positions:

$$\begin{aligned} &s(\pm 1, \pm 1, \pm 1)\\ &s(0, \pm 1/\varphi, \pm \varphi)\\ &s(\pm 1/\varphi, \pm \varphi, 0)\\ &s(\pm \varphi, 0, \pm 1/\varphi) \end{aligned} \tag{4}$$

where s is a scale factor and $\varphi = \left(1+\sqrt{5}\right)/2$ is the so called golden ratio (if a and b are in golden ration, and $a>b$, then $\varphi = \frac{a+b}{a} = \frac{a}{b}$).

Moreover, the Schlegel diagram of the dodecahedron (a planar representation of the 3D solid where the sides never cross) allows us to use a set of five different colors and distribute them around the marker in such a way that, given four visible LEDs, it is possible to instantaneously know the marker rotation. Figure 9a shows the colored dodecahedron Schlegel where a "Y" is highlighted; in Fig. 9b, the actual detection of the "Y" in a captured image, enabling us to know which part of the marker is being seen. With five different colors, there are 20 unique "Y"s, as many as there are in the dodecahedron.

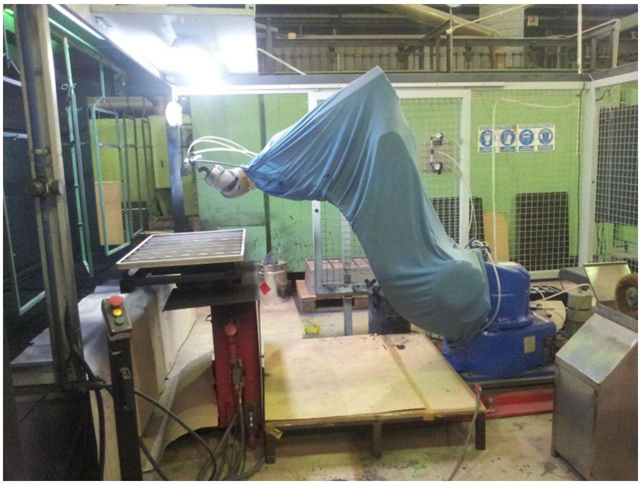


Fig. 14 Motoman EPX2050 paint-specific robot operating in the same area where the human operator performs the demonstration

For the rotation estimation algorithm, the set of known 3D positions is designated by P, where P_i is a 3D vector $[x_i, y_i, z_i]^T$ with the coordinates of the dodecahedron vertex i. P is considered the set of standby positions. From the detected "Y," it is possible to know which LED is which, for all visible LEDs in the images. After retrieving the 3D measures from the stereo analysis, these positions are stored in the matrix Q, the set of measured positions. The nth vector in Q stores is the current world position of the nth stand-by LED in P.

Q and P are said to be paired. To find the marker orientation, we find the rotation from points Q_i to points P_i.

The Kabsch algorithm [31] provides a mean to solve this problem. This method finds the rotation matrix that optimally describes (in a root-mean-squared-error sense) the rotation from two paired 3D point lists:

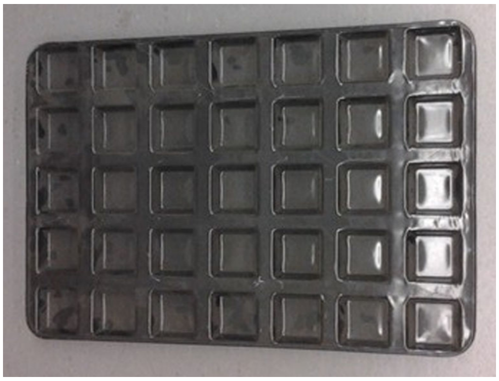


Fig. 15 A common baking tray which is coated at Flupol. The painting skill involves left to right swings of the spray-gun

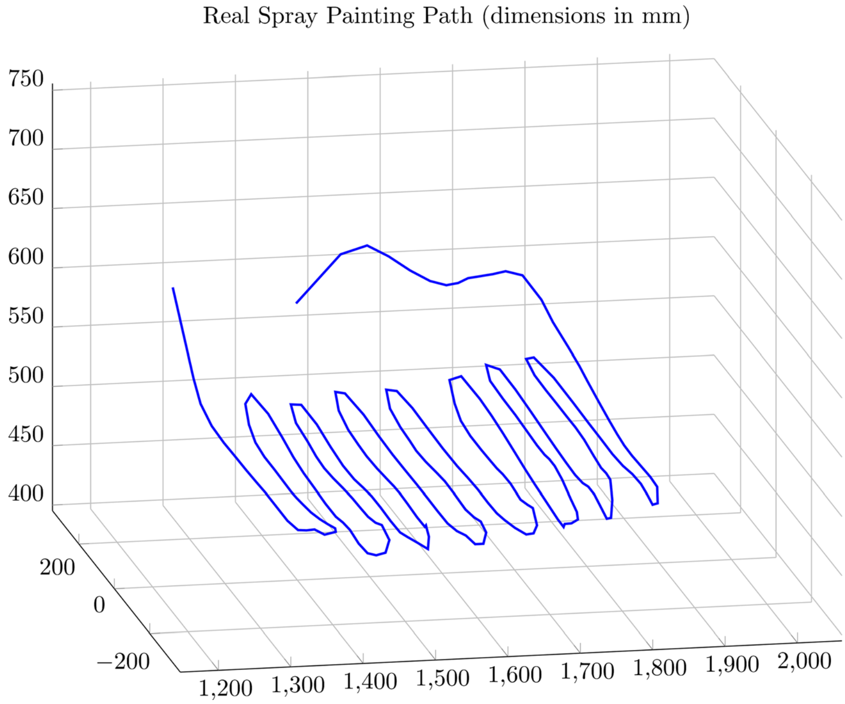


Fig. 16 The captured painting path associated with the part from Fig. 15. This is the final robot path which is a linear interpolation between each way-point

1. P and Q must have origin-centered vectors so the first step is subtracting both sets their respective centroid.
2. Compute the covariance matrix A defined as: $A = P^TQ$.
3. Compute the singular value decomposition of A: $A = USV^T$ and the optimal rotation matrix $W*$ comes from:

$$d = sign\left(det\left(VU^T\right)\right)$$
$$W^* = V\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & d \end{pmatrix} U^T \tag{5}$$

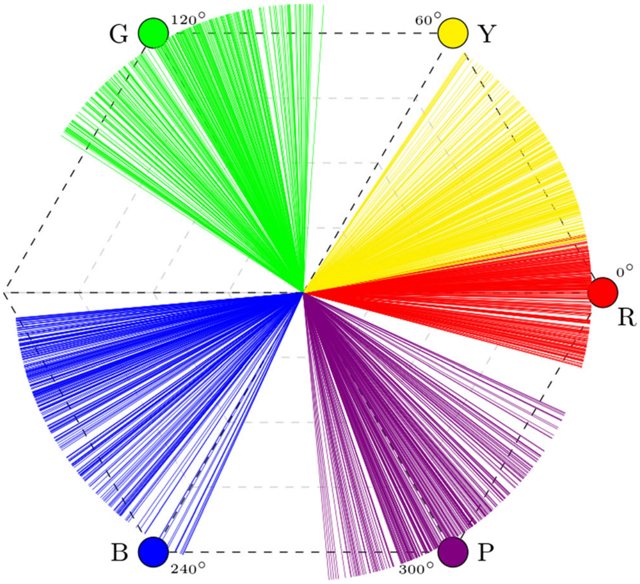


Fig. 17 HSV hue values for a collection of samples from each LED, retrieved after moving the marker on the entire workspace. Almost all color classes are separable except yellow and red

The auxiliary parameter d is used to insure a right-handed coordinated system.

At this point, we have the full (6-DoF) characterization of the movement of the marker (and thus of the work tool), with both position (3-DoF) and orientation (3-DoF). Also note that using only five colors has an added advantage on the detection stage: these five colors are preestablished and the image

Table 1 Number of misclassification of image clusters sorted by color

Image analysis: 100,000 frames

Cluster colour classification

	F/P	F/P
Red	4,544 (4.544 %)	0
Green	0	0
Blue	0	0
Yellow	0	4,544 (4.544 %)
Purple	0	0

F/P are the false positives (another color classified as the current) and F/N are false negatives (the current colour classified as some other)

Table 2 Evaluation of the position and orientation estimation

Stereo frames: 50,000 pairs	
Position estimation	
Success	Failure
50,000 (100 %)	0 (0 %)
Orientation estimation	
Success	Failure
49,919 (>99.8 %)	81 (<0.2 %)
Complete pose estimation	
Success	Failure
>99.8 %	<0.2 %
Average/per demonstration (1,500 frames)<3 fails	

The success and failure rate for each algorithm is presented for a total of 50,000 successfully paired image frames. The complete pose estimation performance is held at the last subtable

processing is focused on finding them while disregarding any extra (noisy) data. Using HSV color space analysis, the five colors can be chosen to have hue values of 0, 60, 120, 240, and 300 as it was used in the examples presented in this paper. In this scenario, these colors represent a linearly separable set of classes that minimizes color misclassification.

Overall, every algorithm used so far is computationally fast and does not compromise the real-time performance of the system.

3 Human-robot interface

The next step after the human movements have been recorded by the motion capture system is sending data to the robot controller that enable the machine to do the exact same gestures. In order to provide a complete abstraction of the programming language to the end-user, a robot program must be automatically generated. It happens that when in possession of a series of time-stamped 6D poses, building a program is straightforward: each pose gives place to a move-linear instruction in robot language to that same position and orientation. The velocity information is extracted by the time stamp of each pose. Figure 10 summarizes the automatic code generation.

At this stage, going from the human skill to the robot mimic is a matter of using a minimal graphical interface (see Fig. 11). It enables the user to start/stop the demonstration process with a simple mouse click. Until the robot program is ready to be uploaded to the controller, no further interaction is needed. The image processing, stereoscopy, pose estimation, and automatic code generation happens instantaneously from the operators' point of view. The time it takes to make a new mouse click in the button<Start Playback>is enough to finish the code generation and upload the program to the robot controller (usually, there is an extra time slot available since the operator (demonstrator) must leave the robot working). The usual offline coding or teach pendant programming are not needed in this scenario. The whole interface resumes to a minimal graphical user interface (GUI) with a couple of buttons to start and stop the demonstration or robot playback.

4 Industrial application: tests and results

The system performance was studied using a single pair of industrial cameras (Fig. 12 (left)) and a marker with 50-mm radius, performing in a real industrial cell for spray coating applications of baking trays (Fig. 12 (right); note the presence of intense overhead lamps to which the tracking system is immune against). All the tests have been performed at the industrial demonstrator: Flupol.

The marker is suited for a range of industrial uses. For the purpose of the aforementioned industrial painting, the marker was attached to a spray painting gun (Fig. 13 (left)). We have also performed tests with an adapted tool with industrial suction cups for metal sheets bending operations (Fig. 13 (right)). In this case, two markers were used to avoid occlusions when handling large metal sheets. This shows well the versatility of the proposed marker.

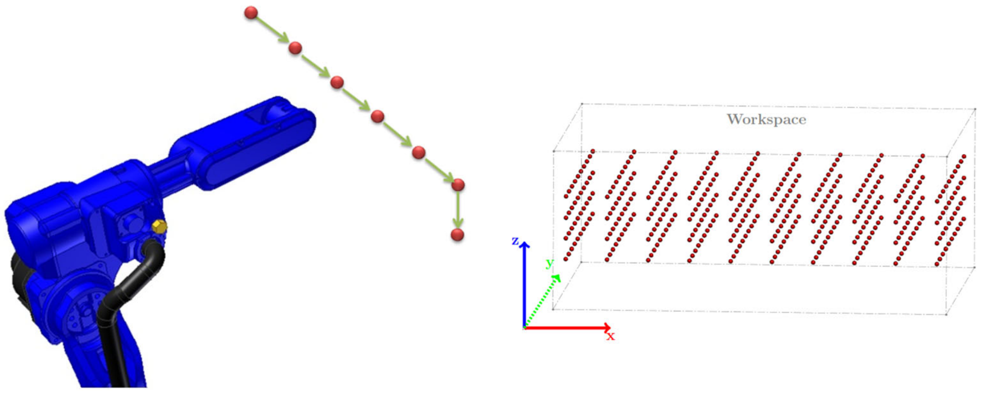


Fig. 18 The manipulator moves in an incremental path and the marker is turned ON at regular intervals (*left*; *red dots*). Care is taken that all positions are visible to both cameras. The final precision grid. After the robot has moved around the entire workspace, the result is a precision grid with a correspondence between tracked positions and robot positions read from the robot controller (*right*)

Table 3 Marker position error on x, y, and z-axes

Maker pose error (X, Y, Z) — (R_x, R_y, R_z) (all distance in mm, angles in degree °)

	x	r	z	R_x	R_y	R_z
Mean error	0.52	−2.1	0.9	0.5	−0.2	−0.5
Mean absolute error	1.1	3.2	1.9	1.0	0.8	1.1
Max. absolute error	3.6	8.2	5.4	4.1	3.9	3.6
Standard deviation	1.5	2.9	1.7	2.2	2.1	1.8

Four measures are presented for each axis: the mean error, the mean absolute error, the maximum absolute error, and the standard deviation

The cameras are from Imaging Source, 1,024×768 CCD sensor with USB 2.0 connectivity. The cameras and the marker are synchronously triggered using a standard microprocessor which is also responsible for reading other interface I/Os and synchronizing them with the video frames. The cameras deliver at 25 fps. The total workspace dimension is (x×y×z) 1,000 mm×1,000 mm×400 mm. The robot is a Motoman EPX2050. It has been mounted on the same workspace where the human operator does the demonstration (check Fig. 14). The robot naturally replaces the human after he has shown the machine how to do the task. There is a complete abstraction of the programming language and no interaction with the teach pendant is required.

For a common baking tray as the one shown in Fig. 15, the spray coating is performed by swinging the gun along the part surface. The corresponding tracked trajectory is presented in Fig. 16; note that the tracked positions are connected by a straight line since the robot program is generated using linear interpolation between the points.

The quality of the robot playback is then dependent of the performance of the proposed marker. First, we show the robustness of the proposed tracking technique. After 100,000 image frames, the cluster color classification according to the HSV analysis is depicted on Fig. 17. It shows how the LEDs are distinguishable, color-wise. When driven properly, the achievable colors by each LED makes the detection very robust even under the lighting conditions presented above in Fig. 12.

Table 4 Marker position error measured as the Euclidean distance from the expected position $\boldsymbol{M}$ to the estimated $\widetilde{M}$

Maker position error: $E = \left\|M - \widetilde{M}\right\|_2$, and orientation error: $\theta = 2\cos^{-1} q_M^{-1} q\widetilde{M}*$ (all distance in mm, angles in degree °)

	E	0
Mean error	3.8	1.7
Max error	8.9	6.2
Std. deviation	2.7	2.1

For orientation, the absolute error is the shortest path (angle) between the two orientations which is computed from the angle between the two corresponding quaternions, q_M and $q\widetilde{M}$

*Possible implementation for the shortest angle between two quaternions

All colors are linearly separable classes in the hue space with exception to yellow and red. Even though, as Table 1 suggests, the misclassification is very low.

Also, it only takes place for one of the colors and there are always another visible LEDs in the image that put the tracking algorithm in the right path. This is demonstrated in Table 2: after 100,000 images, i.e., 50,000 successfully paired image frames, the position estimation (does not depend on color) always succeeds—precision is addressed below; orientation, which depends on the colored Y patterns, has a success rate over 99.8 %. With an average demonstration length of 1,500 frames, it means less than three fails for each run which does not compromise the robot playback.

The precision of the 6-DoF pose tracking was studied using the industrial manipulator itself. To estimate position accuracy, the marker was attached to the robot end effector and this was set to move in a grid pattern, sweeping the entire workspace with a grid spacing of 100 mm as suggested by Figs. 17 and 18.

This renders about 300 positions; at each position, 10 samples were collected. The (robot) coordinates from the test positions are well known and serve as ground truth as they are read from the robot controller. Also, the manipulator was moved to the target positions and remained still for about 4 s to minimize noise from the mechanical vibration. To estimate orientation accuracy, the robot was stopped at five different positions. At each one, the orientation was incrementally changed by 30° for a total of 60 different orientations. For example:

$$R_x = [0^\circ, 30^\circ, 60^\circ] \times R_y = [0^\circ, 30^\circ, 60^\circ, 90^\circ] \times R_z = [-60^\circ, -30^\circ, \ldots, 90^\circ]$$

Table 5 Processing times for every stage of the motion demonstration process

	Routine	Time (ms)	
Online	Image proc. (Debayer and clustering)	2.2	
	2-Camera cluster matching and 3D retrieval	0.5	
	Pose estimation	0.6	
Total			<4
Offline	Robot program generation	30	
	Communication and up-load	1,000	
Total			<1,100

The tests were run on a core i7 @2.8 GHz, under a Linux Mint 12 installation

This depends on how the marker is attached on the robot flange. In the end, there are 300 orientations to evaluate.

Tables 3 and 4 resume the marker pose accuracy; the analysis is done along each direction and also for the Euclidean error (and shortest angle) between the position (and orientation) estimates and the ground truth.

The error pattern is biased; it is especially noticeable along the y direction, which happens to be the depth axis of the stereo configuration. This is due to the biased placement of the cameras; that is, the pair of devices being installed only on one side of the workspace; the sphere fitting algorithm receives samples from just a small area around the sphere—in this case, the left side. Since the cameras are aligned horizontally, which turns to be the robot *x*-axis, the position error tends to be lower along that direction. Concerning orientation, each direction seems equally affected by the estimation noise. The maximum absolute errors take place on the image borders (on the image pane) and at the farthest region from the cameras (in the 3-D world); it is explained by the simultaneous effect of barrel distortion in the cameras (even when compensated during the calibration) and the low stereo performance for far away objects. A second pair of cameras at the opposite side of the first pair capturing the other side of the marker can significantly improve this performance.

Finally, the real-time capability of this system is stated on Table 5. The minimum user interface and fast processing times guarantees that right after the operator gives the order to stop the demonstration process, the robot is ready to perform and mimic the demonstrated movements. Image processing, stereoscopy, and pose estimation are done in real-time. When the demonstration ends and the whole trajectory is available, the robot program is automatically generated. Uploading the program to Motoman NX100 controller may require the major part of the total time. Yet, it varies significantly with each method available (ftp, TCP/IP, etc.) and with the presence of consistency check routines (verifying duplicate names, backups, database integration…). Overall, the average time seems like instantaneous to a common human operator.

5 Global assessment and conclusions

The proposed system presents a novel method for robot programming by demonstration, and has a minimal user interface. It allows factory operators to quickly program an industrial robot by showing it how to move the work tool around the workspace. Despite the simplicity of the computer interface, the system is adequate for industrial use and has a powerful set of key features. It has proven robust to environmental conditions, particularly lighting—no workspace preparation and conditioning is needed. Also, the system is reliable and accurate: tests with a single pair of cameras show an average of 3.8 mm positioning error and 1.7 orientation error; the proposed marker has a pose estimation success rate higher than 99.8 %. Even the remaining 0.2 % do not jeopardize the final playback trajectory executed by the robot. The architecture is flexible enough to adapt to more demanding tasks: more than one marker can be used to prevent occlusions and cameras can be added to improve precision as needed. The markers can be built using a 3D printer and only one pair of industrial cameras is needed, keeping the whole system cost low. The system also performs in real time, making the robot ready to perform as soon as the demonstration ends. There is no intrusion in the task so the workers can operate freely in a casual daily production scenario, without worrying with the robot. One prototype of the system has been installed in a real industrial scenario and used for spray painting applications, with results validated by experienced operators.

Acknowledgments Marcos Ferreira acknowledges FCT—Fundação para a Ciência e a Tecnologia for his PhD grant: SFRH/BD/60221/2009. This work, being part of the Project RL4 COOPERATION—NORTE-07-0124-FEDER000060, is financed by the North Portugal Regional Operational Programme (ON. 2 O Novo Norte), under the National Strategic Reference Framework (NSRF), through the European Regional Development Fund (ERDF), and by national funds, through the Portuguese funding agency, Fundação para a Ciência e a Tecnologia (FCT). This work is also part of Project PRODUTECH PTI (number 13851) - New Processes and Innovative Technologies for the Production Technologies Industry, and has been partly funded by the Incentive System for Technology Research and Development in Companies (SI I&DT), under the Competitive Factors Thematic Operational Programme, of the Portuguese National Strategic Reference Framework, and EU's European Regional Development Fund.

References

1. Ferreira M, Paulo Moreira A, Neto P (2012) A low-cost laser scanning solution for flexible robotic cells: spray coating. Int J Adv Manuf Technol 58(9–12):1031–1041
2. Chen H, Weihua S, Xi N, Song M, Chen Y (2002) Automated robot trajectory planning for spray painting of free-form surfaces in automotive manufacturing. Robotics and Automation, IEEE Int Conf 1: 450–455
3. Chen H, Weihua S, Xi N, Chen Y, Roche A, Dahl J (2003) A general framework for automatic CAD-guided tool planning for surface manufacturing. Robotics and Automation, IEEE Int Conf 3:3504–3509
4. Chen H, Xi N (2008) Automated tool trajectory planning of industrial robots for painting composite surfaces. Int J Adv Manuf Technol 35(7–8):680–696
5. Chen H, Fuhlbrigge T, Li X (2008) Automated industrial robot path planning for spray painting process: A review. Automation Science and Engineering, 2008. CASE 2008. IEEE International Conference on, pages 522–527
6. Norberto Pires J, Godinho T, Ferreira P (2004) CAD interface for automatic robot welding programming. Ind Robot: Int J 31(1):71–76
7. Neto P, Mendes N, Araujo R, Pires JN, Moreira AP (2012) High-level robot programming based on CAD: Dealing with unpredictable environments. Ind Robot 39(3):294–303
8. Neto P, Mendes N (2013) Direct off-line robot programming via a common CAD package. Robot Auton Syst 61(8):896–910

9. Waldherr S, Romero R, Thrun S (2000) A gesture based interface for human-robot interaction. Auton Robot 9:151–173
10. Kumar P, Verma J, Prasad S (2012) Hand Data Glove: A Wearable Real-Time Device for Human-Computer Interaction. Hand, 43
11. Dillmann R, Rogalla O, Ehrenmann M et al. (2000) Learning robot behaviour and skills based on human demonstration and advice: the machine learning paradigm. In Robotics Rresearch, International Symposium, 9, pages 229–238
12. Zollner R, Rogalla O, Dillmann R (2001) Integration of tactile sensors in a programming by demonstration system. Robot Autom, IEEE Int Conf 3:2578–2583
13. Zollner R, Rogalla O, Dillmann R, Zollner M (2002) Understanding users intention: programming fine manipulation tasks by demonstration. Intell Robot Syst, Int Conf 2:1114–1119
14. Rogalla O, Ehrenmann M, Dillmann R (1998) A sensor fusion approach for PbD. IEEE/RSJ Int Conf 2:1040–1045, Intelligent Robots and Systems
15. Hein B, Worn H (2009) Intuitive and model-based on-line programming of industrial robots: New input devices. In Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on, pages 3064–3069
16. Hein B, Hensel M, Worn H (2008) Intuitive and model-based on-line programming of industrial robots: A modular on-line programming environment. In Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on, pages 3952–3957
17. Field M, Pan Z, Stirling D, Naghdy F (2011) Human motion capture sensors and analysis in robotics. Ind Robot 38(2):163–171
18. Elgammal A, Lee C (2009) Tracking people on a torus. Pattern Anal Mach Intell, IEEE Transactions on 31(3):520–538
19. Shon AP, Grochow K, Rao RPN (2005) Robotic imitation from human motion capture using Gaussian processes. In Humanoid Robots, 2005 5th IEEE-RAS International Conference on, pages 129–134
20. Naksuk N, Lee CSG, Rietdyk S (2005) Whole-body human-to-humanoid motion transfer. In Humanoid Robots, 2005 5th IEEE-RAS International Conference on, pages 104–109
21. Sigal L, Balan A, Black MJ (2010) HumanEva: synchronized video and motion capture dataset and baseline algorithm for evaluation of articulated human motion. Int J Comput Vis 87(1–2):4–27
22. Azad P, Asfour T, Dillmann R (2008) Robust real-time stereo-based markerless human motion capture. In Humanoid Robots, 2008. Humanoids 2008. 8th IEEE-RAS International Conference on, pages 700–707
23. Azad P, Ude A, Asfour T, Dillmann R (2007) Stereo-based Markerless Human Motion Capture for Humanoid Robot Systems. In Robotics and Automation, 2007 I.E. International Conference on, pages 3951–3956
24. Cutti AG, Giovanardi A, Rocchi L, Davalli A (2006) A simple test to assess the static and dynamic accuracy of an inertial sensors system for human movement analysis. In Engineering in Medicine and Biology Society, 2006. EMBS'06. 28th Annual International Conference of the IEEE, pages 5912–5915
25. Yamamoto T, Fujinami T (2008) Hierarchical organization of the coordinative structure of the skill of clay kneading. Hum Mov Sci 27(5):812–22
26. Benbasat AY, Paradiso JA (2002) An Inertial Measurement Framework for Gesture Recognition and Applications. In Revised Papers from the International Gesture Workshop on Gesture and Sign Languages in HumanComputer Interaction, GW'01, pages 9–20, London, UK, UK, SpringerVerlag
27. Malheiros P, Costa P, Moreira AP Robust 3D motion capture and object positioning system using light emitting markers synchronized with stereoscopic camera system. UPIN NPat.77/Pat. 41, Int.Patent PCT/IB2009/007186
28. Hartley RI, Zisserman A (2004) Multiple View Geometry in Computer Vision. Cambridge University Press, ISBN: 0521540518, second edition
29. Pratt V, Point P (1987) Direct Least-Squares Fitting of Algebraic Surfaces, Technical report
30. Lukács G, Marshall AD, Martin RR (1997) Geometric least-squares fitting of spheres, cylinders, cones and tori. Technical report
31. Kabsch W (1976) A solution for the best rotation to relate two sets of vectors. Acta Crystallogr A 32(5):922–923