A robot trajectory programming method using multi-camera systems

Silvio Giancola, Davide Chiarion, Remo Sala
Dipartimento di Meccanica, Politecnico di Milano
Via La Masa, 34 - 20158 Milan ITALY

***Abstract*—The paper studies how multi-camera systems can be used for registering trajectories. A sensorized tool (a paintbrush) has been used to track the movements of a painter. Its position and its orientation is recorded using a rigid body registration technique on the reflective spheres, which generates its trajectory. Additional treatments for smoothing the acquired trajectory enable a robot to reproduce its movement. Such technique for the registration of tool movements relieves the robot programmer of a fastidious trajectory planning and remarkably quickens the programming phase.**

I. Introduction

Industrial robots exist from more than 30 years and their programmings were initially very basic [1]. In more recent years, robot programming by demonstration (PbD) has become a central topic of robotics that spans across human robot interaction, machine learning, machine vision and motor control [2]. This paper will focus on machine vision and proposes a robot trajectory planning method based on a multi-camera system.

Classic robot trajectory planning is based on a computer point-by-point programming [3]. It is actually the least time-efficient way to design it because of the complexity to describe trajectory with sampled points. Generally, not all the trajectory points are taught to a robot, because of the fastidiousness of the operation and the redundancy of the transmitted information. Nevertheless, when paths are wealth and movements are complex, a certain precision is required.

Due to trajectory implementation complexity, a few applications never came into being. In order to fasten and facilitate the trajectory programming, we propose a multi-camera system capable of recording the trajectory of a sensorized tool.

The tool, handled by a person, is tracked by a trinocular system that reconstructs the 3D position of known features on a scene. Fixing a set of markers on the tool, this set is projected on the three cameras image plane and 3D reconstruction is provided using trinocular stereoscopy and epipolar geometry concepts. By means of a rigid body registration algorithm, the position and the orientation of the acquired point cloud is estimated, which is to say the tool position. Applying some extra treatments like filters and interpolations, eventual noise is reduced and the trajectory becomes smoother. Passing the acquired trajectory data to a robot, the movement that the person gives to the tool can be duplicated.

This project strives for deeper understanding of pros and cons of using this technique applied in a process of decorating ceramic tiles and objects with a robot. This kind of applications requires a large amount of point describing the trajectory of a brush, that are quite fastidious to define with a classical programming method. The innovative aspect of this work consists in verifying the possibility of obtaining a precise movement, defined and fluid, like a person would produce, as well as accurate and repeatable. Once the actual possibility of the robot to perform smooth motions has been verified through this application, it is possible to request that it simulates other kind of tool actions or even a person's wrist movements.

This document will be articulated around the following sections, firstly the identification of the tool will be explained, then a comparison between smoothing techniques will be provided and an metrological analysis will be performed. Finally, the painter application will be presented as well as some conclusion about this programming method.

II. Tool identification

Multi-camera systems are commonly used for reconstructing three dimensions scenes, when the use of a single camera does not suffice. In this section, a method will be presented for the reconstruction of the position of a tool equipped with a set of spherical markers.

A. Stereoscopic system

Stereoscopic systems are non-intrusive systems using calibrated cameras and exploiting epipolar geometry theory for the reconstruction of points in three dimensions. These points are projected on the camera image planes, paired on the acquired images, then reconstructed on the scene [4]. The most common illustration of stereoscopic systems is the human binocular system, which reconstructs a 3D environment thanks to the two points of view defined by the two eyes.

Reflective spherical markers are objects that are easily detected by cameras, it is sufficient to illuminate the scene and add an adequate filter to each camera to highlight the markers on each image. Also, the projection of the spheres on the cameras image plane is a circle, independently to the point of view.

If a set of reflective markers is integrally fixed around the tool, the stereoscopic system is capable of reconstructing their positions in a scene. Some spheres may be hidden from the field of view of the cameras, but the supernumerary of cameras and markers resolve this occlusion problem.

The epipolar geometry is the fundamental theory of stereoscopy. It introduces the concept of epipolar plane, defined by

978-1-4799-2280-2/14/$31.00 ©2014 IEEE

the two centers of the cameras C_l and C_r and the object point M, as illustrated in figure 1. Also, epipolar lines are defined by the intersections of the epipolar plane with the sensors. The first camera defines a 3D line of solutions for the object point position and the second camera reduces the solutions to a single point. Further lectures for stereoscopic reconstruction is available in [4].

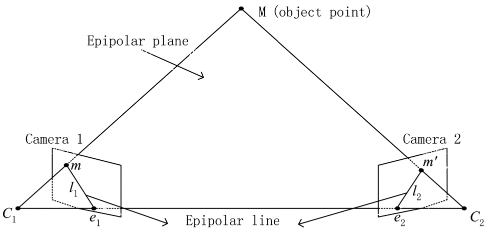


Fig. 1. Epipolar geometry with two cameras

Stereoscopy systems based on two cameras can be used for the reconstruction of a single point but it becomes equivocal for a set; the closeness of many epipolar planes cannot permit the correct coupling of homogeneous points.

For avoiding such results, a trinocular vision system is used [5]. Having a third camera provides us two secant epipolar lines for every marker on every sensor for a simpler, more reliable and more accurate 3D reconstruction.

Algorithmically, the method is the following:

- A first point $P_{1,i}$ is selected on the 1^{st} image. Its epipolar plane is constructed with the 2^{nd} camera and the corresponding epipolar line is projected on the 2^{nd} camera image.
- The closest point $P_{2,j}$ to the drawn line is selected. Its epipolar plane is constructed with the 3^{rd} camera and the corresponding epipolar line is projected on the 3^{rd} camera image.
- Once again, the closest point $P_{3,k}$ is found, then its epipolar plane is constructed with the 1^{st} camera and the corresponding epipolar line is projected on the 1^{st} camera image.
  - If the closest point respect to the last epipolar line corresponds to the first point $P_{1,i}$, the loop is closed and the three points $P_{1,i}$, $P_{2,j}$ and $P_{3,k}$ are the projections of the same 3D points. A consistent reconstruction can be done.
  - If the loop is not closed, the algorithm tries to match the second closest point to every straight line, until the correspondence is found.

In that way, homogeneous points are associated using a "depth first" tree algorithm, using "back tracking" technique for searching the solutions [6]. Given that the points are sorted regarding the distance to the corresponding epipolar line, the algorithm is time-efficient because the first tries are the more probable.

Note that the cameras need to be calibrated: a first step consists in the correction of the lens distortions and in the determination of intern orientation; then a successive step regards the finding of the relative positions of the 3 cameras respect to the world coordinate system. Many techniques exist for retrieving these parameters. The one used is based on a Zhang technique, using a pattern recognition for the definition of the world reference system, more informations can be found in [7], [8].

B. Point Cloud Indexing

A set of 3D points scattered is an environment is usually called a point cloud. Our trinocular system is able to retrieve the 3D position of the point cloud defined by the set of markers fixed on the tool. In order to successively treat this point cloud and to extract the geometry of the tool from it, each point must be consistently indexed.

To correctly sort the point cloud, information about the relative distances between every point are used. Acquired points are randomly indexed and the corresponding matrix of distances is created. A matrix of distances gathers the information of the distances between all the point in the cloud (see equation 3). In fact, the value of the element d_{ij} of the matrix, positioned on the the i^{th} row and the j^{th} column corresponds to the distance between the i^{th} and the j^{th} points of the cloud.

$$Distances = \begin{bmatrix} d_{11} & d_{12} & \dots & d_{1n} \\ d_{21} & d_{22} & \dots & d_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ d_{n1} & d_{n2} & \dots & d_{nn} \end{bmatrix} \tag{1}$$

Defining a model for the matrix of distances, we refer the points in a certain order. For successively treating of the point cloud, we need to index our acquired point cloud as the model. It is thus suitable to catch eventual inversion or loss of markers.

It has been identified that in case of inversion of two markers, an inversion of both column and row indexes occurs (see equation 2a). In case of loss of a marker because of occlusion, the row and the column indexes are filled with wrong distances (see equation 2b).

$$\begin{bmatrix} 0 & d_{13} & d_{12} & d_{14} \\ d_{31} & 0 & d_{32} & d_{34} \\ d_{21} & d_{23} & 0 & d_{24} \\ d_{41} & d_{43} & d_{42} & 0 \end{bmatrix} \qquad \begin{bmatrix} 0 & d_{12} & X & d_{14} \\ d_{21} & 0 & X & d_{24} \\ X & X & 0 & X \\ d_{41} & d_{42} & X & 0 \end{bmatrix} \tag{2}$$

(a) Inversion of 2^{nd} and 3^{rd} marker

(b) Occlusion of 3^{rd} marker

The implemented algorithm for catching the inversion or the loss of data is the following:

- If n is the number of marker, we split the $n \times n$ matrix in n rows. Be row_i the i^{th} row.
- For each row_i, we define an empty $2 \times n$ array named ind_i, linked to the i^{th} row.
- For each element d_{ij} of row_i, its value is matched to a correspondent value $\hat{d}_{kl}$ of the model. The rows and

columns indexes (k and l) are then saved in the columns of ind_i.

- For each ind_i, we define i_{sol} the largest occurrence of an index in this matrix. If all the element of ind_i is 0, then a marker is missing. Else, the largest occurrence of an index different to 0 returns the correct index of i.

Example: I know I should find 5 markers, but one was not reconstructed (occlusion) and points are indexed randomly. The matrix of distances is:

$$Distances = \begin{bmatrix} 0 & d_{13} & d_{15} & d_{12} & 0 \\ d_{31} & 0 & d_{35} & d_{32} & 0 \\ d_{51} & d_{53} & 0 & d_{52} & 0 \\ d_{21} & d_{23} & d_{25} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{3}$$

from which the 5 index matrices are extracted:
$ind_1 = \left(\begin{smallmatrix} 0 & 1 & 1 & 1 & 0 \\ 0 & 3 & 5 & 2 & 0 \end{smallmatrix}\right)$, *the* 1^{st} *marker corresponds to the* 1^{st} *one*
$ind_2 = \left(\begin{smallmatrix} 3 & 0 & 3 & 3 & 0 \\ 1 & 0 & 5 & 2 & 0 \end{smallmatrix}\right)$, *the* 2^{nd} *marker corresponds to the* 3^{rd} *one*
$ind_3 = \left(\begin{smallmatrix} 5 & 5 & 0 & 5 & 0 \\ 1 & 3 & 0 & 2 & 0 \end{smallmatrix}\right)$, *the* 3^{rd} *marker corresponds to the* 5^{th} *one*
$ind_4 = \left(\begin{smallmatrix} 2 & 2 & 2 & 0 & 0 \\ 1 & 3 & 5 & 0 & 0 \end{smallmatrix}\right)$, *the* 4^{th} *marker corresponds to the* 2^{nd} *one*
$ind_5 = \left(\begin{smallmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{smallmatrix}\right)$, *a marker is missing (the* 4^{th}*).*

C. Rigid Body Registration

The rigid body registration is a process based on similarities. Similarities are geometric transformations composed of a rotation, a translation and a scale factor. Rigid body registration techniques are used for finding the similarity parameters between two point clouds (see figure 2).

The rigid body registration is applied on the acquired point cloud rigidly fixed to the tool. The model point cloud is the tool set in a reference position, where the tool coordinate system coincides with the reference coordinate system.

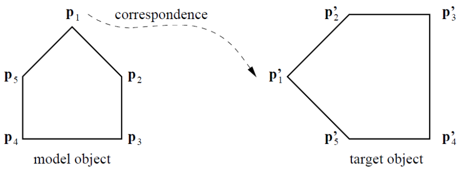


Fig. 2. Rigid body registration

One of many techniques of rigid body registration is based on singular value decomposition (SVD). The algorithm is the following:

- *1st step:* be $\{p_i\}_{i=1..n}$ the n points of the model and $\{p'_i\}_{i=1..n}$ the n measured points, ordered as the model. Both centroids of the clouds are respectively noted $\hat{p}$ and $\hat{p}'$, then r_i and r'_i are the vectors from relative centroids to the points.
- *2nd step:* be s the scale factor, defined by the equation 4.

$$s = \frac{\sum_{i=1}^{n} \|r'_i\|^2}{\sum_{i=1}^{n} \|r_i\|^2} \tag{4}$$

- *3rd step:* be M a matrix defined by $M = \sum_{i=1}^{n} r'_i {*} r_i^T$. Be U, S, and V the three matrices of the SVD decomposition of M. Be R the rotation matrix, defined by the equation 5.

$$R = U * \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & sign(det(U * V^T)) \end{bmatrix} * V^T \tag{5}$$

- *4th step:* be T the translation vector, obtained by the equation 6.

$$T = \hat{p}' - s * R * \hat{p} \tag{6}$$

It is mathematically shown that the SVD method for the rigid registration converges on a solution for s, R and T that minimizes the fitting error with a least squares method. Further lecture for rigid registration is available in [9].

Actually, for our reconstruction case, the scale factor should return 1. It is possible in the 4^{th} step to impose exactly $s = 1$ for a more coherent reconstruction.

With our stereoscopic system, we have been able to acquire the position of rigid point fixed on a tool, which were then used for extracting the position of the tool in the environment. Tracking the tool, we can queue its positions in function of the time, and thus create trajectories.

III. Trajectory smoothing

In order to study the trajectory acquired by the stereoscopic system, a model has been established. The acquired trajectory is decomposed in three translations and three rotations time-variant functions. Each of these continuous functions is characterized by a certain variations related to the dynamism of the trajectory, and present a certain noise representing the uncertainty of the instrument. Figure 3 shows an example of acquired trajectory component.

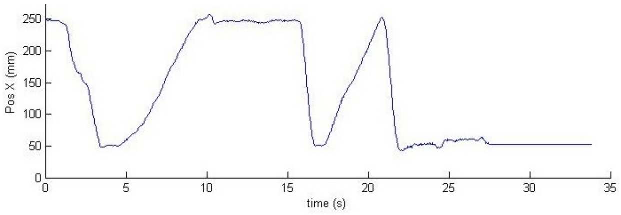


Fig. 3. Example of a component of the trajectory, expressed as time-variant function to smooth

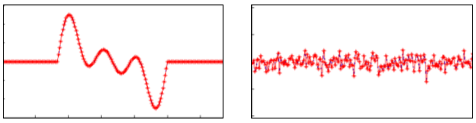
These functions represent the evolution of the trajectory in time. We establish that these rough curves are a combination of basic time-variant sinusoidal functions, on which white Gaussian noise is added. In order to obtain a consistent trajectory reproducible by a robot, these functions need to be cleaned from the instrument noise, without distorting the movement.

In this section, we will present different methods for cleaning and smoothing trajectories. A first category of method for smoothing trajectory is the interpolations. *Cubic spline interpolation*, *Bezier curves* and *NURBS fitting* have been taken into account [10]. A second category regards filters where *low-pass filters* will be presented.

A. Spline Interpolation

Splines interpolation uses cubic polynomial for linking sampled points between each other. Exploiting the polynomial degrees of freedom, the velocities and the accelerations are set continuous. Because of acceleration continuity, spline interpolation results to be efficient for the strain of the robot.

In figure 4 are shown the effect of spline interpolation on a combination of sinusoidal functions and on a pure white Gaussian noise function. The blue curves is the theoretical function and the red one the interpolated curve. Regrettably, this smoothing technique does not filter the noise at all.



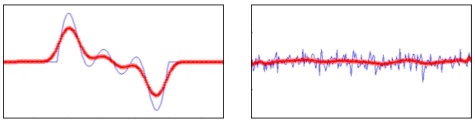
(a) Sinus combination (b) White Gaussian noise

Fig. 4. Spline interpolation on different trajectory models

B. Bezier Curves

Bezier curves is a type of curves defined by a series of controlling points. Using the acquired points, it is possible to obtain a Bezier curve that smooths the trajectory. Algorithmically, the Bezier curve is obtained using the Casteljau algorithm [10].

Figure 5 illustrates the effect of Bezier curve interpolation on the same previous fundamental functions. This interpolation approaches the acquired points, but controlling points do not belong to the curve. Thus, Bezier curves interpolation are able to perfectly clean all the noise, but they also reduce the amplitude of a movement. Trajectories are too smoothed and are not consistent respect to real ones.



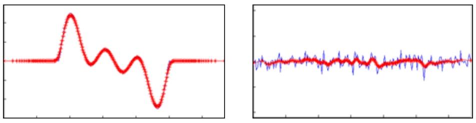
(a) Sinus combination (b) White Gaussian noise

Fig. 5. Bezier curve interpolation on different trajectory models

C. NURBS Fitting

Non Uniform Rational B-Spline (NURBS) fitting is a generalization of Bezier curves [10]. NURBS fitting introduces a degree of freedom that allows the tuning of the fitting.

The figure 6 illustrates the effect of a smoothing with a 50^{th} degree NURBS on the same previous fundamental functions.



(a) Sinus combination (b) White Gaussian noise

Fig. 6. NURBS fitting on different trajectory models

The function dynamism is saved and the white Gaussian noise in relatively reduced. Nevertheless, the NURBS fitting provides a variable re-sampling of the data, which reduces the number of information.

D. Low-Pass Filtering

Low pass filtering is a different type of trajectory smoothing based on frequency study. Figure 7 reveals the frequency components of the function given in the introduction example (figure 3). The spectrum is composed of low frequencies defining the actual movement, and high frequencies characterized as noisy.

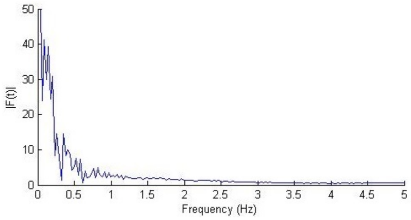
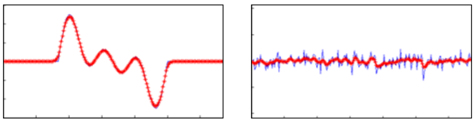


Fig. 7. Amplitude Spectrum of an acquisition

Building a Low-Pass filter permits to reduce the high frequency noise and focus on the actual movement. The filter cut off is a parameter for tuning the level of smoothness in function of the application.

The figure 8 illustrates the effect of a smoothing with a Low-Pass filter with a 3 Hz cut off frequency, on a combination of sinusoidal functions and on a pure white Gaussian noise function. The function dynamism is saved and a part of the white Gaussian noise in reduced.



(a) Sinus combination (b) White Gaussian noise

Fig. 8. Low-Pass filtering on different trajectory models

E. Results

Spline interpolation is not able to clean a white Gaussian noise, Bezier curves cleans the noise but doesn't reconstruct consistent trajectories and NURBS fitting re-samples data, thus changes the resolution of the trajectories.

In the end, Low-Pass filtering allows a correct smoothing of trajectories. It reduces efficiently the high frequency noise and maintains consistency in the smoothed trajectories. Moreover, it is parameterizable with a cut off frequency. This method provides the setting of a level of dynamism that wants to be kept. The more dynamic the actual trajectory is, the higher this cut off frequency must be for smoothing consistency.

The fact that Low-Pass filtering smooths our trajectories and can be tuned in function of the application makes this technique an useful instrument.

IV. Metrologic Analysis

The Guide to the expression of Uncertainty in Measurement [11] splits the uncertainty in two components which are accuracy and repeatability. The accuracy represents the difference between the mean of the acquisition in respect to the target value, meanwhile the repeatability represents the standard deviation of the set in respect to its mean. This section describes the uncertainty of the trinocular vision system.

A. Single marker examination

A study has been done on blob detection method for the evaluation of pixel projection on the camera sensor. For the measurement of the marker on the image, a "SmallestCircle" algorithm has been chosen, which fits the area of the projected marker with the smallest circle. 200 measures of the center of this smallest circle have been realized, with a 10 mm diameter marker placed on different distances and different positions of the field of view.

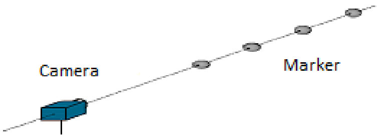


Fig. 9. Setup for single marker measurement examination

Uncertainty of the measure has been plotted on figure 9. Standard deviations have been measured in function of the projected circle area of the marker on the sensor. The different colors represent different positions of the blob on the sensor.

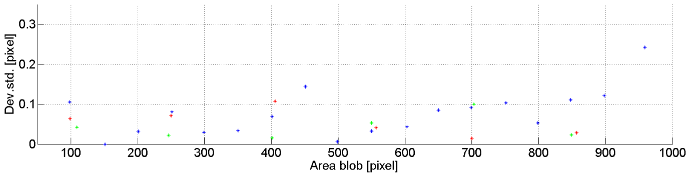


Fig. 10. Repeatibility of blob detection for a single marker)

The global uncertainty of the single marker examination by the camera sensor is equal to 0.15 pixel.

B. Monte Carlo method

In order to reconstruct the overall measurement chain, and define the tool position measurement uncertainty, a Monte-Carlo method has been applied. Monte-Carlo method estimates the output uncertainty of a system or a technique using the input uncertainty values of the system and realizing a large amount of measurements [9].

The system taken into account is the trinocular vision system which returns the position of a tool. In input, it has the positions of the markers projected on the camera image planes and in output the position of the tool. In the previous section, we defined a 0.15 pixel input repeatability. Realizing a large amount of a static tool position measurements, we can identify the uncertainty.

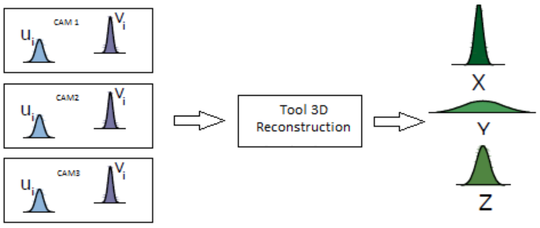


Fig. 11. Monte Carlo Method for uncertainty definition

Notice that the tool position uncertainty depends on the application, on how the camera are set and on the marker are fixed on the tool. We will apply the Monte Carlo method and calculate the uncertainty for our specific application.

V. Application : A Painting Process

The trajectory recording technique has been tested on a concrete application, a painting process. Industrial painting process is typically an application were robot programming is complex to implement with traditional techniques. A painter produces brush trajectories that are not rigorously defined but that are the result of an artistic process. Using our system for planning a robot trajectory should facilitate and fasten the programming.

Two setups have been used for this application. The first setup consists on a painting movements *recorder*, composed of the stereoscopic system, an object to paint and a set of sensorized paintbrush. The second setup consists on a painting movement *duplicator*, composed of a robotic cell, the same object to paint and the same set of paintbrush adapted at the robot gripping ability.

A. The acquisition setup

The registration setup is composed of three CCD cameras making up the trinocular vision system, with a 1280 × 1024 pixels resolution and a 12 mm lens, all synchronized with a 30 Hz trigger for a regular acquisition. The positions of the cameras have to minimize the reconstruction error, and the set must be able to reconstruct the paintbrush even if data from one camera is missing. This is why the cameras should be the most distant possible one from each other, preferably covering the three directions of the three axis reference frame.

The painting brushes are sensorized with four reflective spheres, set up for having different distances between each others. Light-weight carbon bars are used for fixing the reflective spheres, minimizing in that way the impact on the global weight. They are placed on an unused part of the brush, avoiding a fastidious taking for the artist.

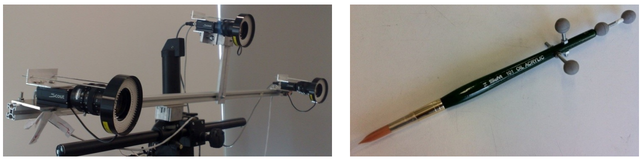


Fig. 12. Trinocular vision system and sensorized tool

Circular infrared lights with a wavelength of 850 nm are mounted on each camera and appropriate filters leak the light reflected by the markers. The camera are singularly calibrated, then registered one respect to a common coordinate system fixed on the object to paint.

B. The robotic cell

The robotic cell is composed of an ABB IRB 120, which is a small and dynamic anthropomorphic robot with a reachability of 580 mm, a payload of 3 kg and an arm load of 0.3 kg. It is a fast robot, with an accuracy in its position of 0.02 mm and a repeatability of 0.01 mm. The paintbrushes are the same than the ones used for the recording layout, and are fixed on the robot flange.

The robot coordinate system is also set on the object to paint, similarly to the acquisition setup. For doing so, you need 2 similar setup from which you can define either the vision coordinate system than the robot coordinate system.

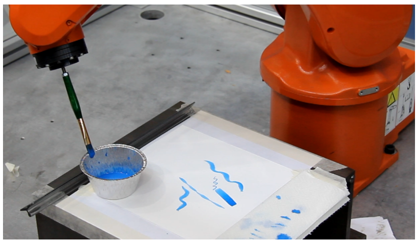
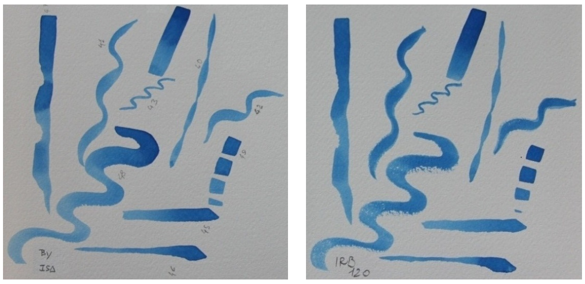


Fig. 13. Robot cell

C. Results

The smoothing technique used is a low-pass filter with a cut off frequency of 10 Hz. Painting reproductions are presented in figure 14. Two different paintbrushes were used for the tries, a sharp one and an flat one. In both cases, the movement is quite well reproduced.



(a) Painter result (b) Robot result

Fig. 14. Comparaison between registration and reproduction painting

Further metrologic analysis have been done. The overall system constructs a trajectory with a theoretical precision of *0.26 mm* using the previously calculated uncertainty of the camera in the Monte-Carlo simulation. The system is supposed to be repeatable.

However, the system presents some systematic errors. The *yielding of the brush tip* decreases the reproducibility of the system. In fact, the form of the paintbrush is slightly different whether it is dry, wet or full of paint.

Successively, the *overlap between the robot and the vision coordinate systems* must be accurately done, else a systematic error will be present all the trajectories. The more similar the layouts are, the more repeatable the technique is.

A last aspect that comes into play is the *position of the markers* on the sensorized tool. Their positions need to be known accurately respect to the tip of the paint brush.

For these reasons, the uncertainty in the duplication grow around the millimeter.

VI. Conclusions

We have studied how multi-camera systems can be used for registering movement. The different aspects of the duplication have been studied, from the data acquisition to the trajectory reproduction. The tool has to be sensorized and the trajectory smoothing adapted in function of the application and the required dynamism of the movement to record.

The use of stereoscopic vision is not common in industrial applications, but this method presents interesting performances for programming complex trajectories more easily. This method has been applied in a concrete industrial painting process, returning an accuracy around the millimeter.

The principal limitation of this method is the setup required. When traditional programing methods require only a computer, this method need additional equipments. Furthermore, the robot is not creative by himself, it only reproduces an already registered movement.

Anyway, this technique can be an interesting tool for a few applications that were hardly projected using previous and traditional offline programming techniques. This method can be applied in many applications, for the robot reproduction of painting, welding or assembling movements, but also more basically in trajectory registration for objects tracking.

References

[1] T. Lozano-Perez, “Robot programming,” *Proceedings of the IEEE*, vol. 71, no. 7, pp. 821–841, 1983.

[2] S. Calinon, “Robot programming by demonstration,” in *Springer handbook of robotics*. Springer, 2008, pp. 1371–1394.

[3] H. Giberti, S. Cinquemani, and S. Ambrosetti, “5r 2dof parallel kinematic manipulator–a multidisciplinary test case in mechatronics,” *Mechatronics*, vol. 23, no. 8, pp. 949–959, 2013.

[4] R. Hartley and A. Zisserman, *Multiple view geometry in computer vision*. Cambridge university press, 2003.

[5] N. Ayache and F. Lustman, “Trinocular stereo vision for robotics,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 13, no. 1, pp. 73–85, 1991.

[6] S. J. Russell, P. Norvig, J. F. Canny, J. M. Malik, and D. D. Edwards, *Artificial intelligence: a modern approach*. Prentice hall Englewood Cliffs, 1995, vol. 2.

[7] Z. Zhang, “A flexible new technique for camera calibration,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 22, no. 11, pp. 1330–1334, 2000.

[8] G. MAINETTI, “Calibrazione di telecamere per sistemi di visione stereoscopica: Confronto tra algoritmi genetici e tecniche tradizionali,” 2011.

[9] W. H. Press, *Numerical recipes 3rd edition: The art of scientific computing*. Cambridge university press, 2007.

[10] L. Piegl and W. Tiller, “The nurbs book. 1997.”

[11] *Guide to the Expression of Uncertainty in Measurement*. International Standard Organization, Geneva (Switzerland), 1993.