A Vision-based Correction of Inertial Measurement of Human Motion for Robot Programming by Demonstration

Robin Pellois and Olivier Brüls
Department of Aerospace and Mechanical Engineering,
Faculty of Applied Science, University of Liège, Belgium
Email: {robin.pellois, o.bruls}@uliege.be

***Abstract*—We propose in this work an original approach consisting in correcting inertial human hand trajectory with vision-based object tracking in a context of programming by demonstration (PbD) of pick-and-place tasks. One challenge in PbD is to record human demonstrations accurately enough, in an easy way which does not limit human motion. Merging inertial-based and vision-based technologies may take advantage of both and fulfill the requirement of a PbD process. Our method is based on the identification of Positions of Interest (POIs) from object and hand data, corresponding to picking or placing actions. Then objects POIs are paired with hand POIs to modify the human hand trajectory. The method is implemented on a Sawyer robot with Xsens IMU sensors. Pick-and-place tasks with different complexity have been recorded and reproduced by the robot. The robot succeeds to reproduce the demonstrated tasks which validates our method.**

***Index Terms*—programming by demonstration, vision, inertial human motion tracking, merging**

I. Introduction

One challenge in PbD [1] is to acquire human demonstration that is meaningful for the robot, i.e., managing the correspondence issue. With kinesthetic method [2], the correspondence issue is trivial but limit human motion. The challenge is then to develop a human demonstration acquiring method which gathers the intuitiveness and the easiness-to-use that PbD yearns, with a sufficient level of accuracy and which does not limit human motion.

Field et al. [3] expose a survey of human motion capture methods in robotics. Each technology has its own advantages and drawbacks. A merging solution can potentially benefit from the advantages of both. Some solutions have already been proposed to merge vision-based and inertial-based systems. For instance, in [4], an accurate system to track tool trajectory for industrial painting is propounded. These works reach a suitable level of accuracy for a PbD process but require complex equipment. This work targets easier implementation.

According to [5], human motion measurement in industrial environment requires to be corrected. In addition, environment observations are assumed to be less complex than human motion measurement. We therefore explore the approach of correcting the inertial human motion measurement with environment observation. The inertial-based human motion estimation method used in this work was developed in a previous work. As a first step, we develop this approach in the context of pick-and-place task. In such a context, the environment observation consists in tracking object position. As shown in [6], [7] numerous tools for object detection in robotics are available and do not require heavy equipment, which fulfills the requirements of a PbD process.

The method is detailed in section II. Section III presents the implementation of the method and the experimental results. The conclusions of the study are summarized in section IV.

A. Inertial Human Hand Trajectory Estimation

The inertial human hand trajectory tracking method used in this work is based on the estimation of the orientation of each segment (arm, forearm and hand). The IMU orientation estimation only relies on accelerometer and gyroscope data. The mean error is between 28.5 mm and 61.8 mm.

B. Clustering Method

The choice for clustering method should take into account that the data present outliers and noise, the number of clusters is not known a priori and not all the data have to be involved in any cluster. For these reasons, partitional clustering methods such as the K-means algorithm [8] and its derivatives are discarded. According to [9], hierarchical clustering methods are less sensitive to the initial condition, they are robust to noise and there is no need to specify the number of clusters in advance. However, these algorithms are computationally expensive. According to [10], so-called density-based algorithms isolate the dense areas that are separated by sparse areas. These methods are thus well suited for our problem and we opt in particular for DBSCAN (Density-Based Spatial Clustering of Applications with Noise) [11]. This algorithm creates clusters from core points identification. A point is an n-component vector. A core point is a point

This work was supported by the Interreg Greater Region program under the Robotix Academy Project.
Manuscript received January 21, 2022; revised April 11, 2022.

doi: 10.18178/ijmerr.11.6.411-416

with at least $MinPts$ points within distance ϵ. The two parameters of the algorithm are then ϵ and $MinPts$.

C. *Matching Method*

The problem of matching points is largely documented in computer vision application involving 2D points, 3D points, or spatiotemporal points. In [12], a descriptor is associated to each key point based on its surroundings pixels. A putative set of pairs is created and the challenge consists in eliminating the non-desired pairs. This approach can also be seen in [13] which compares different algorithms to eliminate false pairs (RANSAC, K-nearest-neighbour graph, Graph Transformation Matching). In [14], the matching is solved by using the Haussdorff distance to quantify the similarity between 2 different sets of points. A large variety of methods exists depending on the application and the final goal. Therefore, we propose here a specific method for our application. A difficulty in our context is to identify the exact subset of hand and object POIs that should be paired. Once the subsets are identified, the matching problem is reduced to an assignment problem. Our approach is based on an iterative process consisting of identifying roughly the transformation between the hand and the object POIs and eliminating too distant POIs. Identifying the transformation is a point set registration (PSR) problem largely documented [15]. As both dataset contains outliers POIs and noise, the kernel correlation based method [16] is selected for our application.

II. Method

The method is based on the identification and pairing of POIs from object and inertial data. A POI $\mathcal{P}$, defined as a stationary position during the task, is a 5-component vector. It consists of a 3D position P and two temporal components indicating the beginning s and the end e of the POI: $\mathcal{P}=\left[s,P,e\right]$. It is considered that the time interval of two POIs from the same trajectory (hand or object trajectory) cannot overlap.

A. *Time-space Scaling*

For clustering and point matching, it is necessary to compare data in space and time which have different physical units. So, numerical data need to be scaled. Let us consider a set S of N points P_i with $i \in [1,N]$, where each point is a vector of n temporal and spatial components. The scaled set $s=[p_1,\dots,p_N]$ is composed of scaled points corresponding to the z-score of the original data:

$$G=\frac{1}{N}\sum_{i=1}^{N}P_i \tag{1}$$

$$p_i=[\frac{P_i(1)-G(1)}{\sigma(1)}\dots\frac{P_i(N)-G(N)}{\sigma(N)}] \tag{2}$$

With σ the standard deviation of the original data. Scaled data are then adimensional.

B. *POIs Identification*

The object data recorded from the camera and the hand position estimated from IMUs data present different features. IMUs give a constant flow of data at 100 Hz but which suffer from noise, measurement errors and estimation errors. On the other hand, occultation of the camera and computational image treatment make the flow of data discontinuous. However, camera data are accurate and present a lower noise. Furthermore, objects are more static than human hand during the demonstration. Therefore, the DBSCAN algorithm is applied differently to each dataset.

1) Object POIs

It is considered that an object tracking method has been applied to the data from the camera recording the task. Let us consider that N objects are identified. For each object j ($j \in [1,N]$), its trajectory Γ^j is recorded. The trajectory Γ^j, consisted of N^j 3D positions P_i and times t_i with $i \in [1,N^j]$, is noted:

$$\Gamma^j=[\,(t_1\;\;P_1)\dots(t_i\;\;P_i)\dots(t_{N^j}\;\;P_{N^j})\,]^T. \tag{3}$$

Each trajectory Γ^j is scaled following the method presented in II-A and noted γ^j. G and σ are computed from the combination of the hand and object data. For each γ^j, the DBSCAN algorithm is applied and returns n_c clusters. A cluster C_k ($k \in [1,n_c]$) is made of n_k points of γ^j:

$$C_k=[\,(t_u\;\;P_u)\dots(t_v\;\;P_v)\,] \tag{4}$$

A POI $\mathcal{P}_k$, associated to the cluster C_k, is computed as

$$\mathcal{P}_k=[\min([t_u\dots t_v]),\frac{1}{N_k}\sum_{j=u}^{v}P_j,\max([t_u\dots t_v])]. \tag{5}$$

The DBSCAN algorithm is applied on the scaled data but the computation of the POIs is made on the original values. The set of extracted POIs depends on the value of the DBSCAN algorithm parameters. A too high value of ϵ could miss some POIs and a too low value would create unnecessary POIs. For a better robustness, we opt for a low value of ϵ and add a post treatment to merge POIs that are close to each other. To do so, two POIs, $\mathcal{P}_j$ and $\mathcal{P}_k$, temporarily close: $s_k<e_j+t_{limit}$; or spatially close: $||P_k-P_j||<d_{limit}$; are merged as

$$\mathcal{P}_j=[\min([s_k,s_j]),\frac{P_k+P_j}{2},\max([e_k,e_j])]. \tag{6}$$

Finally, POIs $\mathcal{P}_j$ with a too short time interval $e_j-s_j<t_{limit}$ are eliminated.

When an object is manipulated several times, the intermediate POIs have to be split. In addition, object POIs associated with non-manipulated objects are discarded. A non-manipulated object is an object which has only one POI lasting the complete demonstration.

2) Hand POIs

The continuous flow of data from IMUs makes the DSCAN algorithm particularly sensitive to its parameters ϵ and $MinPts$ and may then detect unwanted POIs or omit others. For a better robustness, the DBSCAN algorithm is applied on the scaled human hand trajectory N_ϵ times with different values of ϵ : $E=[\epsilon_1,...,\epsilon_{N_\epsilon}]$, $MinPts$ is fixed. Hand data are scaled similarly than object data. To each value ϵ_i ($i\in[1,N_\epsilon]$), a set of potential POIs $\mathcal{P}^i$ is formed from the output of the DBSCAN algorithm, no post treatment is made. Then, the complete set $\mathcal{P}$ of all potential POIs is built:

$$\mathcal{P}=[\mathcal{P}^1 \ldots \mathcal{P}^{N_\epsilon}] \tag{7}$$

In a second step, $\mathcal{P}$ is scaled and noted p which is then used as input for the DBSCAN algorithm. The set $\mathcal{P}$ of 5-component vectors is used to compute G and ϵ for scaling. The clustering is then less sensitive to the parameters values. The n POIs $[\mathcal{P}_u \ldots \mathcal{P}_v]$ belonging to the same cluster k are merged to create the final POI $\mathcal{H}_k$ as

$$\mathcal{H}_k=[\min([s_u \ldots s_v]),\frac{1}{n}\sum_{k=u}^{v}P_k,\max([e_u \ldots e_v])]. \tag{8}$$

Afterwards, a post treatment is applied. First, the POIs from the set $\mathcal{H}$ are sorted chronologically according to their start time. Secondly, the POIs are fused or separated according to their spatiotemporal proximity. Let us consider two consecutive POIs $\mathcal{H}_i$ and $\mathcal{H}_{i+1}$ temporally close i.e., $e_i > s_{i+1}-t_{limit}$. If they are also spatially close $||H_i - H_{i+1}|| < d_{limit}$, both POIs are merged as $\mathcal{H}_i=[\min(s_i,s_{i+1}),\frac{H_i+H_{i+1}}{2},\max(e_i,e_{i+1})]$. If 2 POIs are distant, i.e., $||H_i - H_{i+1}|| > d_{limit}$, the POIs are separated. To separate two POIs, two cases are distinguished. The first case occurs when is temporally included in $\mathcal{H}_i$. In that case only e_i is modified as $e_i = s_{i+1}-t_{limit}$ to prevent from suppressing the included POI. The second case occurs if the time intervals of $\mathcal{H}_i$ and $\mathcal{H}_{i+1}$ are overlapping, i.e. $e_i < e_{i+1}$. Then, e_i and s_{i+1} are modified proportionally to their POIs time interval as

$$\alpha=\frac{e_i-s_i}{(e_i-s_i)+(e_{i+1}-s_{i+1})} \tag{9}$$

$$\mathcal{H}_i=[s_i\,,H_i\,,e_i-\alpha(e_i-s_{i+1})-\frac{t_{limit}}{2}] \tag{10}$$

$$\mathcal{H}_j=[s_{i+1}+(1-\alpha)(e_i-s_{i+1})+\frac{t_{limit}}{2}\,,H_{i+1}\,,e_{i+1}]. \tag{11}$$

This post treatment procedure is applied to the complete set $\mathcal{H}$ several times until no POIs can be suppressed. Finally, POIs with time interval lower than t_{limit} are discarded.

C. Pairing of Hand and Object POIs

Let us consider a set $\mathcal{H}$ of hand POIs and a set $\mathcal{O}$ of object POIs, respectively containing N_H and N_O POIs. The following step consists of identifying the set $\mathfrak{P}$ of N_p pairs. A pair $\mathfrak{P}_i$ with $i\in[1,N_p]$ contains a hand and an object POIs: $\mathfrak{P}_i=[\mathcal{H}_j\,,\mathcal{O}_k]$ with $j\in[1,N_H]$ and $k\in[1,N_O]$. A POI can be part of only one pair. The aim of this part is to identify the right set of pairs $\mathfrak{P}$.

The first step of this procedure consists of identifying the subsets of hand and object POIs which should be involved in the pairs. The second step creates the pairs by assignment based on matching time.

1) Identification of POIs to pair

This procedure is iterative. At each step, either a hand POI or an object POI is discarded until a certain condition, explained later, is reached.

First, the spatial transformation between hand and object POIs is estimated. The error of orientation of the inertial hand trajectory being mainly around the z-axis, the transformation is composed of a rotation $R_z(\theta)$ of an angle θ around the z-axis and the translation $T=[T_x\;T_y\;T_z]^T$. The estimation is made through the minimization of the cost function [16] with a Gaussian kernel correlation function KC :

$$COST(O,H,\theta,T)=\sum_{i=1}^{N_O}\sum_{j=1}^{N_H}-KC(O_i,R_z(\theta)H_j+T) \tag{12}$$

$$KC(O_i\,,H_j)=(2\pi\sigma^2)^{-N/2}\exp(\frac{-||O_i-H_j||^2}{2\sigma^2}). \tag{13}$$

Then, the transformed hand POIs set $\mathcal{H}'$ is computed by modifying the spatial parts as $H'_j=R_z(\theta)H_j+T$.

Secondly, unwanted POI is eliminated. The two sets of POIs $\mathcal{H}'$ and $\mathcal{O}$ are scaled as h' and o . G and σ used for scaling are computed from the dataset gathering $\mathcal{H}'$ and $\mathcal{O}$. Let us introduce the distance function $d(o_i,h_j{}')$ as

$$D1=||[s_i\,,o_i]-[s_j\,,h'_j]|| \tag{14}$$

$$D2=||[e_i\,,o_i]-[s_j\,,h'_j]|| \tag{15}$$

$$D3=||[e_i\,,o_i]-[e_j\,,h'_j]|| \tag{16}$$

$$D4=||[s_i\,,o_i]-[e_j\,,h'_j]|| \tag{17}$$

$$d(o_i,h_{j'})=\min(D1,D2,D3,D4) \tag{18}$$

An appropriate distance function is necessary due to the differences between hand and object POIs time interval.

Using the distance d and considering a hand POI h_j', the $d_{mean}(h_j')$ is computed as

$$d_{mean}(h_j') = \frac{1}{N_O} \sum_{i=1}^{N_O} d(o_i, h_j') \qquad (19)$$

A hand POI with a d_{mean} value higher than k_d is a candidate to elimination. Among all the candidates, only the POI with the highest d_{mean} value is discarded during the current step of the iterative process. When no more hand POIs are candidates to elimination and the two remaining subsets of hand and objects POIs have the same size, the iterative process stops. If the two subsets do not contain the same number of POIs, the process continues according to two cases. If $N_H > N_O$, the hand POI with the highest d_{mean} value is discarded (even if lower than k_d). If $N_O > N_H$, for each remaining object POI o_i, the d_{mean} value is computed similarly:

$$d_{mean}(o_i) = \frac{1}{N_h} \sum_{j=1}^{N_h} d(o_i, h_j') \qquad (20)$$

The object POI with the highest d_{mean} value is discarded.

2) Assignement of hand and object POIs

Since the object and hand POIs to pair are identified, the problem is reduced to an assignment problem. In this work, we used the Hungarian method [18] to solve it based on a cost matrix. The cost matrix M of size $N_H \times N_O$ (or $N_p \times N_p$) represents the temporal distance between POIs and is computed as

$$M_{i,j} = d^t(\mathcal{H}_i, \mathcal{O}_j). \qquad (21)$$

with d^t the temporal distance function between 2 POIs

$$D1 = \| s_i - s_j \| \qquad (22)$$

$$D2 = \| e_i - s_j \| \qquad (23)$$

$$D3 = \| e_i - e_j \| \qquad (24)$$

$$D4 = \| s_i - e_j \| \qquad (25)$$

$$d^t(\mathcal{H}_i, \mathcal{O}_j) = \min(D1, D2, D3, D4). \qquad (26)$$

D. Correcting Trajectory

1) Global correction

The first step consists in applying a transformation to the trajectory. The transformation consists of a rotation R and a translation T and is estimated through the minimization of the following function

$$e(R,T) = \sum_{k=1}^{N_p} O_k - (R H_k + T) \qquad (27)$$

With N_p the number of pairs. Then the transformation is applied to the complete human hand trajectory. A local correction of the trajectory is then applied.

2) Local correction

The inertial human hand trajectory Γ is sequenced in K parts Γ_k according to all the hand POIs $\mathcal{H}$ detected initially. The trajectory Γ alternates between hand POIs and motion part Γ_k. Let us Γ_k be the part of the trajectory from the hand POIs $\mathcal{H}_u$ to $\mathcal{H}_a$ with n points in between:

$$\Gamma_k = [(e_u H_u),(t_1 p_1),(t_2 p_2)...(t_{n-1} p_{n-1}),(t_n p_n),(s_a H_a)]^T . \qquad (28)$$

Then the hand POIs are replaced by the object POIs according to the pairing $\mathfrak{P}$. Considering the POI $\mathcal{H}_u$ belonging to the pair $\mathfrak{P}_w = (\mathcal{H}_u, \mathcal{O}_v)$ and the POI $\mathcal{H}_a$ belonging to the pair $\mathfrak{P}_c = (\mathcal{H}_a, \mathcal{O}_b)$, Γ_k is modified as

$$\Gamma_k = [(e_u O_v),(t_1 p_1),(t_2 p_2)...(t_{n-1} p_{n-1}),(t_n p_n),(s_a O_b)]^T . \qquad (29)$$

To avoid abrupt motion between O_v and p_1, and between p_n and O_b, f and g points are discarded:

$$\begin{aligned}\Gamma_k = [&(e_u O_v),(t_{f+1} p_{f+1}),(t_{f+2} p_{f+2}) \ ... \\ &(t_{n-g-1} p_{n-g-1}),(t_{n-g} p_{n-g}),(s_a O_b)]^T .\end{aligned} \qquad (30)$$

The f and g values are proportional to n and computed as $f = \lfloor k_f \times n \rfloor$ and $g = \lfloor k_g \times n \rfloor$ with k_f and k_g two gains lower than one. Finally, an interpolation is made on the motion part Γ_k to create a smooth trajectory I_k. The modified trajectory Γ^{mod} is then the sequence of the interpolated motion part and POIS.

II. Implementation of the Method

A. Experimental Setup

The method is applied on a Sawyer robot from Rethink Robotic. A gun-shaped tool is used to reduce potentially disturbing fingers motion (see Fig. I a)).

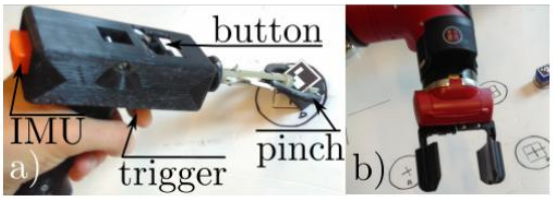

Figure 1. a) gun-shaped tool, b) robot end-effector

This tool receives an IMU for hand orientation tracking and a button activated by a trigger to communicate with the robot during the process. The trigger also commands a pinch at the tip of the tool to manipulate objects during the demonstration. The IMUs used in this work are Xsens MTw Awinda sensors [18].

The objects are 30 mm side cubes with an ArUco marker on a face to easily track their position with the library *openCV* [19]. The camera used in this work is the embedded black and white camera at the wrist of the Sawyer robot, its resolution is 752 x 480 pixels. The camera being directly part of the robot, objects positions with respect to the robot base frame are easily computed. The error on object position has been estimated between 0.1 mm and 18.8 mm. The tasks executed by the operator are described in Table I. After the demonstration of the task by the operator, the objects are put back to their initial position. Then the robot executes the task again: once by following the hand trajectory from IMU measurement only and a second time by following the corrected trajectory. The task is considered a success if the footprint of the objects covers a part of their targeted final position. Such a tolerance is justified by the level of inaccuracy of the object detection method.

For object POIs detection, ϵ and $MinPts$ are respectively tuned to 0.02 and 2. For the first step of the hand POIs detection, E = [0.010, 0.012, 0.014, 0.016, 0.018, 0.020, 0.022, 0.024, 0.026, 0.028, 0.030] and $MinPts$ is tuned to 10. For the second step, ε is tuned to 0.02 and $MinPts$ to 2. The t_{limit} parameter is tuned to 0.5 s and d_{limit} to 55 mm. The optimization process is conducted by the *scipy.optimize* library. The threshold k_d is tuned to 3. Finally, the two gains k_f and k_g are both arbitrary chosen to 0.30. All parameter values have been tuned by trial-and-error on experimental data. However, it can be mentioned that a part of the method is applied on scaled data leading to use similar values (the ϵ values for object and hand POIs detection are both tuned to 0.02 and E have values around 0.02). The $MinPts$ parameter values depend mostly on the frequency of acquisition of data. Object position method presents a frequency of acquisition around 10 Hz and the IMUs, 100 Hz. The values experimentally found are consistent with the frequency of acquisition. The d_{limit} parameter value depends on the quality of the data. The value used in this experiment is within the error range for object or hand position. The parameter d_{limit} is tuned to 0.5 s.

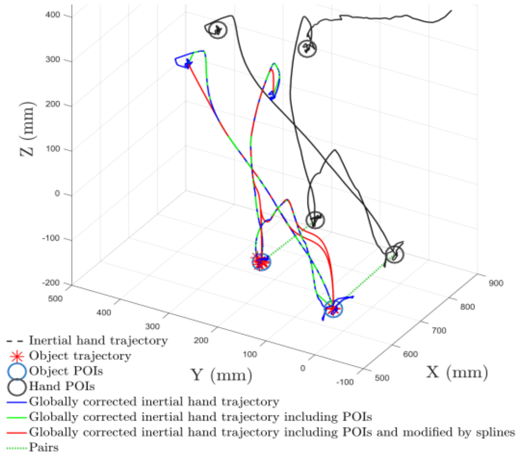

Figure 2. Modified trajectory for task 1

B. Results

The results are presented in Table I. Fig. II shows the experimental results for task 1. The last 2 columns represent the success ("1") or failure ("0") of the robot reproducing the demonstrations, first with non-corrected trajectory ("IMU only"), then with the corrected trajectory. The results show that the inertial method by itself is not accurate enough for the robot to reproduce the task. With the correction from environment observation, each task has been reproduced correctly.

TABLE I. TASKS DESCRIPTION AND EXPERIMENTAL RESULTS

	Tasks properties			Experimental results				
	obj on table	obj moved once	obj moved twice	N_H	N_O	N_P	IMU only	Vision-based corrected trajectory
1	1	1	0	4	2	2	0	1
2	2	1	0	4	3	2	0	1
3	2	2	0	7	4	4	0	1
4	3	0	1	7	4	4	0	1
5	3	2	0	8	5	4	0	1
6	3	2	1	13	8	8	0	1
7	4	2	0	8	6	4	0	1
8	4	4	0	14	8	8	0	1
9	4	3	1	16	10	10	0	1
10	4	2	2	19	12	12	0	1

III. CONCLUSIONS

This work develops a novel approach for merging inertial-based measurement of human hand motion and vision-based measurement of object position for an accurate estimation of the human hand trajectory. It is used as method for acquiring demonstration for programming by demonstration of a pick-and-place task. The method consists in 3 steps: extracting POIs (positions of interest), pairing POIs and modifying the inertial measurement of the human trajectory. The method has been tested with different pick-and-place tasks. The results validate the suitability of the method in the context of programming by demonstration. As future work, the robustness of POIs detection could be increased for instance by investigating the use of different data such as gripper.

CONFLICT OF INTEREST

The authors declare no conflict of interest.

AUTHOR CONTRIBUTIONS

Robin Pellois and Olivier Brüls jointly developed the method. The experimental tests have been conducted by Robin Pellois. The paper has been written by Robin Pellois and reviewed by Olivier Brüls.

REFERENCES

[1] A. Billard, S. Calinon, R. Dillmann, and S. Schaal, "Robot programing by demonstration," *Springer Handbook of Robotics*, Springer, 2008, pp. 1371–1394.

[2] C. Breazeal, M. Berlin, A. Brooks, J. Gray, and A. L. Thomaz, "Using perspective taking to learn from ambiguous demonstrations," *Robotics and Autonomous Systems*, vol. 54, no. 5, pp. 385–393, 2006.

[3] M. Field, D. Stirling, F. Naghdy, and Z. Pan, "Motion capture in robotics review," in *Proc. IEEE Int. Conf. Control Autom.*, pp.

1697–1702, 2009.
[4] V. H. Pinto, A. Amorim, L. Rocha, and A. P. Moreira, "Enhanced performance real-time industrial robot programming by demonstration using stereoscopic vision and an IMU sensor," in *Proc. IEEE Int. Conf. Auton. Robot Syst. Compet.*, 2020.
[5] T. Choi, C. Park, H. Do, D. Park, J. Kyung, and G. Chung, "Trajectory correction based on shape peculiarity in direct teaching manipulator," *Int. J. Control. Autom. Syst.*, 2013.
[6] Y. He and S. Chen, "Advances in sensing and processing methods for three-dimensional robot vision," *Int. J. Adv. Robot. Syst.*, 2018.
[7] D. A. Dewi, E. Sundararajan, A. S. Prabuwono, and L. M. Cheng, "Object detection without color feature: Case study autonomous Robot," *Int. J. Mech. Eng. Robot. Res.*, vol. 8, no. 4, pp. 646–650, 2019.
[8] E. Forgy, "Cluster analysis of multivariate data: Efficiency versus interpretability of classification," *Biometrics*, 1965.
[9] M. G. H. Omran, A. P. Engelbrecht, and A. Salman, "An overview of clustering methods," *Intell. Data Anal.*, 2007.
[10] A. Amini, T. Y. Wah, and H. Saboohi, "On density-based data streams clustering algorithms: A survey," *J. Comput. Sci. Technol.*, 2014.
[11] M. Ester, H. P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," in *Proc. Second International Conference on Knowledge Discovery and Data Mining*, 1996.
[12] D. Wedge, D. Huynh, and P. Kovesi, "Using space-time interest points for video sequence synchronization," in *Proc. Conf. Mach. Vis. Appl.*, 2007.
[13] X. J. Shan and P. Tang, "A robust false matching points detection method for remote sensing image registration," *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.*, 2015.
[14] D. P. Huttenlocher, G. A. Klanderman, and W. J. Rucklidge, "Comparing images using the Hausdorff distance," *IEE Trans. Pattern Anal. Mach. Intell.*, 1993.
[15] H. Zhu *et al.*, "A review of point set registration: From pairwise registration to groupwise registration," *Sensors*, 2019.
[16] Y. Tsin and T. Kanade, "A correlation-based approach to robust point set registration," *Eur. Conf. Comput. Vis.*, 2004.
[17] H. W. Kuhn, "The Hungarian method for the assignment problem," *Nav. Res. Logist. Q.*, 1955.
[18] M. Paulich, M. Schepers, N. Rudigkeit, and G. Bellusci, "Xsens MTw Awinda: Miniature wireless inertial-magnetic motion tracker for highly accurate 3D kinematic applications," *Whitepaper*, 2013.
[19] S. Garrido-Jurado, R. Muñoz-Salinas, F. J. Madrid-Cuevas, and M. J. Marín-Jiménez, "Automatic generation and detection of highly reliable fiducial markers under occlusion," *Pattern Recognit.*, vol. 47, no. 6, pp. 2280–2292, 2014.

Robin Pellois is a PhD candidate at the University of Liège. He received a master's degree from the Ecole de Technologie Supérieure in Montreal (QC, Canada) and from the engineering school Arts et Métiers Paritech (France). His area of research includes robotics and human motion.

Olivier Brüls is a professor at the Department of Aerospace and Mecanical Engineering in the University of Liège. After a master's degree obtained in 2001, he achieved a doctorate at the University of Liège in 2005. Since 2008, he is in charge of the Mecatronic and Multibody Systems Laboratory at the University of Liège. His research concerns mechanical system dynamics, mecatronique, numerical simulations, control and optimization with applications in robotics and biomechanics.