HAL
open science

Recognition of Technical Gestures for Human-Robot Collaboration in Factories

Eva Coupeté, Fabien Moutarde, Sotiris Manitsaris, Olivier Hugues

To cite this version:

Eva Coupeté, Fabien Moutarde, Sotiris Manitsaris, Olivier Hugues. Recognition of Technical Gestures for Human-Robot Collaboration in Factories. The Ninth International Conference on Advances in Computer-Human Interactions, Apr 2016, Venise, Italy. hal-01306482

HAL Id: hal-01306482

https://hal.science/hal-01306482

Submitted on 24 Apr 2016

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

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

Recognition of Technical Gestures for Human-Robot Collaboration in Factories

Eva Coupeté, Fabien Moutarde, Sotiris Manitsaris and Olivier Hugues
Center of Robotics, Mines ParisTech
PSL Research University
Email: first-name.last-name@mines-paristech.fr

***Abstract*—Enabling smooth Human-Robot collaboration requires enhancing perception and intelligence of robots, so that they can "understand" the actions performed by the humans with whom they are interacting. In this paper we are dealing with new industrial collaborative robots on assembly-line and supply-chain in automotive manufacturing. We are conducting research on technical gestures recognition, to allow the robot to understand which task is being executed by human worker, and react accordingly. We use two kinds of sensors: depth-camera for monitoring of human movements, and inertial sensors placed on tools. In this study, we propose and use a method for head and hands tracking using a top-view depth-map, and use HMM (Hidden Markov Models) to recognize gestures with these data. Then, we refine the results from the HMM with data from inertial sensors equipping tools. Our research shows that: i) using 3D-vision only, we can obtain already good results of gestures recognition for several workers: 80% of the gestures are correctly recognized, ii) exploiting data from tools equipped with inertial sensors significantly improve the recognition accuracy to 94% in the same multi-user evaluation. A first test of our method with a simple Human-Robot collaboration scenario is also described.**

Keywords–Human-robot collaboration; Industrial application; Assembly line; Gestures recognition; Depth camera.

I. Introduction

Robots are becoming more and more present in our everyday life. They can be used for social interaction or for medical support. In the industrial context, collaborative robots are emerging that are intrinsically "safe". These robots, devoted to tasks that are either of low added-value, or potential source of musculoskeletal disorders, are working nearby workers without barriers between them contrary to current robots in factories. Therefore, collaborative robots allow increased automation of factories, saving of space and cost while improving productivity in the industrial plants. This new configuration of collaboration between robots and humans on assembly-line and supply-chain is efficient only if human-robot collaboration can be smooth, i.e., the robot is following the human gestures in order to respond fluidly. To this end, and in order to ensure workers' safety, a collaborative robot has to be aware its environment, to be able to adapt its speed to the worker rapidity, and monitor worker's actions in order to ensure smooth cooperation.

Gesture recognition can meet these needs: by recognizing the worker's gestures, the robot can recognise which task is being executed, adapt its speed and detect when something unexpected happens. One of the difficulties of this goal is that, contrary to most Human-Computer interactions where the user can adapt to the system, the worker must be able to work "as usal" and is not supposed to make any effort for his gestures to be correctly understood by the robot.

In this paper, we are focusing on technical gestures recognition for human-robot collaboration. To enable a robust gesture recognition, we use two kinds of sensors: depth-camera and inertial sensors. Depth-camera allows us to have information on the worker's postures, and to evaluate his distance to the robot. In addition, we equip with inertial sensor some tools manipulated by human workers, in order to know when they are used. The main contribution of our paper is to show the feasibility of a robust user-independent and totally non-intrusive system for real-time recognition of typical technical gestures of human workers in a factory, for allowing natural interaction of collaborative robots. Also, one of the novelties in our work is the use, as a complement to real-time analysis of depth images, of an inertial sensor placed on a tool rather than on the operator. This increases very significantly the rate of correct gesture recognition, while keeping our system totally non-intrusive.

This paper is organized in five parts. In Section II, we present related work on human-robot interaction, gesture recognition, human pose estimation, and sensors fusion. In Section III, we present our experimental use-case. In Section IV, we detail our methodology to recognize the gestures and to fuse data from the depth-camera and the inertial sensors. In Section V, we present our results of gesture recognition and a first Human-Robot collaboration using our system. Finally in Section VI, we provide conclusions and perspectives for our future work.

II. Related work

In this Section, we present related work on the topics of human-robot collaboration and gestures recognition. In subsection II-A, we present several applications of human-robot collaboration in different fields. Then in II-B, we describe different kinds of features and frameworks for gestures recognition with depth camera only and with the addition of inertial sensors.

A. Human-Robot collaboration

With the massive arrival of robot in our everyday life the research on human-robot interaction and collaboration has been very active these last years [1].

Robots are already very present in different fields. They are used to interact with elderly people, [2], to guide visitors in a museum [3] or to assist humans during walking training [4]. To be better accepted by humans, the interaction with the robot has to be natural, using the same codes as a Human-Human collaboration. Since gestures are important factors in Human-Human conversations, non verbal interaction has already been

Copyright (c) IARIA, 2016. ISBN: 978-1-61208-468-8

used to enable a communication between the robot and the human, [5] and [6].

In factories, new collaborative robots are designed to be intrinsicly safe and to provide complementary skills to human co-workers like the Kuka LWR [7] and the Universal Robot UR [8]. In this context, collaborative robot and human can work together, for example carrying the same object [9]. Some studies have evaluated the worker's acceptability to work with this new kind of partner [10]. In [11], the authors present a framework enabling the robot to adapt his behaviour to a new user after a first step of joint-action demonstration.

B. Human pose estimation and gesture recognition

Human pose estimation is becoming easier since the apparition of depth-cameras, for example the Microsoft Kinect [12], [13]. This RGB-D sensor combines standard RGB images with depth information obtained by analysis of a projected infrared pattern. We can sort the methods to estimate a human pose using a depth-camera in two classes: without, [14], and with, [15] and [16], a prior posture learning. The advantage of the first class is that we do not need a previous dataset to establish the human pose but these methods can be slower to compute the human pose. Conversely, with prior learning the acquisition of a dataset and its processing can be a long first step, but the human pose estimation is performed more quickly. However, a limitation of human pose estimation using learning is that it can properly handle only postures similar to those included in the dataset used for its design and training. For example the Kinect SDK recognizes postures only when the user is facing the camera. Once these postures are extracted, they can be selected, fully or partially, to determine which ones are the most discriminant to recognize gestures. In [17], the authors use Principal Component Analysis (PCA) to select the most useful features, among skeleton joints, to recognize gestures. In [18], the authors convert the set of skeleton joints to create a more robust and less redundant pose descriptor based on joint-angles representation. They also use SVM to determine key poses which will be used to do gesture training and recognition.

Many methods to recognize gestures have been set up these past years. The 3D skeleton tracking of the subject with a depth camera is often used to recognize gesture, as in [19] and [20]. Various approaches have been proposed to handle dynamic gestures recognition. The most known are HMM [21] used, among other, in [22] and [23]. But other machine-learning algorithms have been successfully applied: Dynamic Time Wrapping (DTW), Support Vector Machines (SVM), decision forest and k-Nearest Neighbours (kNN). All gesture recognition systems have limits and constraints, for example the HMM based recognition needs a large amount of training samples to be efficient. For all systems, the number of recognizable gestures is predefined and limited. To be well distinguished, two gestures must be different enough to enable the recognition system to differentiate them.

The simultaneous utilisation of inertial sensor and depth-camera have appeared in the literature in first place to deal with calibration issue, as in [24]. Fusion of depth-map and inertial sensor for gesture recognition is done in [25]. The authors fused the data coming from the two types of sensor by re-sampling and filtering. Then, they proceed at a classification with HMM. In [26], the authors used only inertial sensor to recognize technical gestures. However, up to our knowledge no published work proposed, as we do in this paper, to improve gestures recognition by using inertial sensors placed not on monitored human, but rather on object or tool manipulated by him.

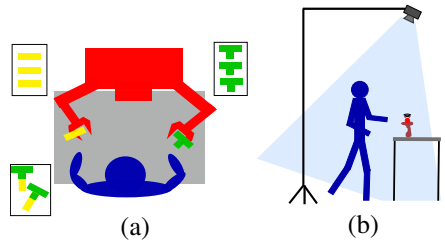

Figure 1. Description of our experimental use-case, (a): the robot gives motor parts to the worker, (b): we equipped the scene with a depth-camera and an inertial sensor fixed on a screwing-gun tool

III. Presentation of our experimental use-case

We work on a scenario where the worker and the robot share the same space and work together. The task is inspired from the assembly of motor hoses on supply chain. Presently, the assembly process of motor hoses has some drawbacks: the worker has to find the appropriate parts of the motor hoses among other motor parts, which is a lack of time and increase the cognitive load of the worker. In our set-up, the robot and the worker are facing each other, a table is separating them, see Figure 1(a).

On an assembly line, because mounting operations must be achieved quickly and efficiently, the actions to be executed by human operators are standardized as a rather strictly-defined succession of elementary sub-tasks. To ensure a natural human-robot collaboration, the robot has to perform an action accordingly to which task the operator is executing, in order to be useful at the right time and not delay the worker. In our use-case, the assembling of motor hoses requires the worker to take 2 hose parts respectively on left and right side, join them, screw them, take a third part from left, join it, screw it, and finally place the mounted motor hose in a box. The only actions performed by the robot are giving a piece with the right claw and giving a piece in the left claw. The set of human operator's gestures to be recognized by our system is therefore rather straightforwardly deduced from above-mentioned sub-tasks as:

1) to take a motor hose part in the robot right claw (G1)
2) to take a motor hose part in the robot left claw (G2)
3) to join two parts of the motor hose (G3)
4) to screw (G4)
5) to put the final motor hose in a box (G5)

These gestures will allow the robot to be synchronized with the operator by understanding when an operator is taking a piece from a claw and when the next piece is needed.

The classical sequence of gestures to assemble motor hoses is: (G1 then G2) or (G2 then G1), then G3, then G4 then G2, then G3, then G4, then G5. Some workers prefer to do the two screwings after the second execution of G3, so that we cannot suppose a strictly-defined ordering of operations, as it is essential to leave to human workers some freedom degree in their work.

We equipped the scene with a depth-camera which is filming the worker with a top-view. With this set-up, we are avoiding most of the possible occultations on a supply-chain due to workers or objects passages. We also put an inertial sensor on the screwing-gun, in order to know when it is moved by the worker, see Figure 1(b).

IV. METHODOLOGY

In this Section, we present our methodology to achieve technical gestures recognition. The global processing pipeline is illustrated on Figure 2: using segmentation and computation of geodesics on top-view depth image, we estimate a 10D feature vector characterizing operator's posture, and used for recognition of gestures. In subsection IV-A, we present our method to extract hands positions of the worker from the depth-map using geodesic distances between the top of the head and each point of the upper body of the worker. In subsection IV-B we explain how we put together information from the depth-map about the worker posture to create features to learn and recognize technical gestures. In subsection IV-C we present our learning and recognition framework, and in IV-D we show our method to merge information from the depth-map and the inertial sensors.

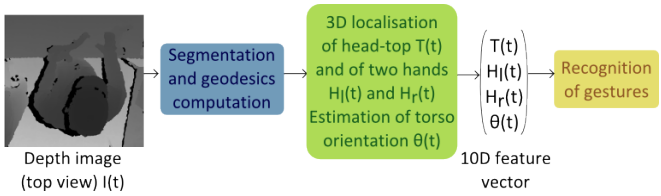

Figure 2. Pipeline from our raw depth image to gestures' recognition

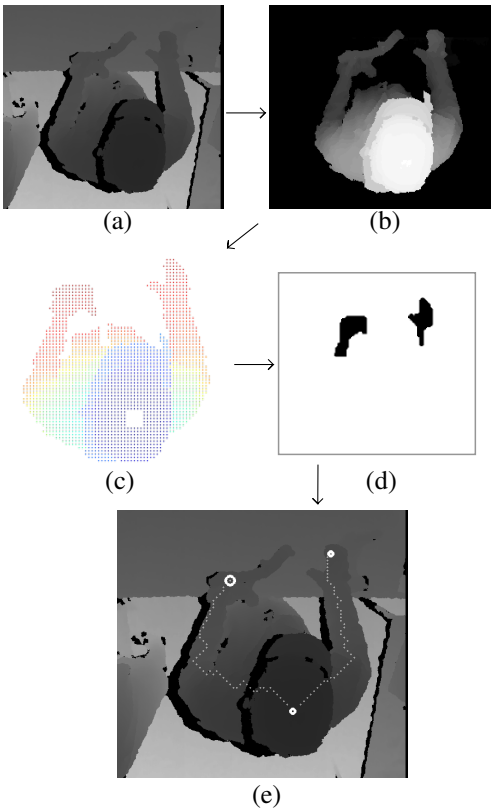

Figure 3. Our hands-tracking method. (a): initial depth map, (b): upper body of the worker, (c): geodesic distances between each point of the upper body and the head (blue to red), (d): farthest upper body parts from the head, (e): head and hands locations with the shortest paths between the hands and the head

A. Tracking hands positions

To localize and track hands of the worker, we have adapted and extended to top-view case the method proposed in [14]. Figure 3 illustrates our framework.

1) Upper-body and head extraction: We only need the upper-body of the operator to recognize his gestures, because the worker is staying in small area during his work and he only uses his hands for assembling the motor hoses. From the raw depth-image (see Figure 3(a)), we segment the upper-body by keeping only the depth pixels that are above the assembling table (see typical result on Figure 3(b)). Then, the top of the head is located as the biggest "blob" from the 10% highest pixels of upper-body (the 10% threshold value was determined by anatomic considerations). Finally, the center of the head top is estimated as the center of the "top-of-head blob". Additionally, using typical anatomic values of height distance from head-top to shoulders level, we locate the two shoulders on the depth-image map; this allows to estimate torso orientation as the horizontal angle of the straight line joining the two shoulders.

2) Creation of the upper-body graph: In order to locate hands, we make the assumption that they are the visible parts of the upper-body that are the farthest from the head, not in Euclidean straight line distance, but following the body surface. To find these "farthest" points, we calculate the geodesic distances between head-top and all points of the upper-body. In order to compute geodesics, we create a weighted graph with all pixels from the upper-body as nodes. The first step is to connect each pixel with its eight neighbours (only if they also belong to the upper-body). To each of these connections is associated a weight equal to the absolute value of difference between the two connected pixels (which is approximately their height difference, since the depth camera is pointing downwards and nearly vertically). In order to obtain geodesics that link only anatomically contiguous body parts (e.g. for instance, not "jumping", from an arm directly to the torso it touches), we prevent connections between depth pixels that have a too large depth difference.

3) Application of the Dijkstra algorithm for hands localization: After creation of the upper-body graph, we apply to it the Dijkstra [27] algorithm in order to find the shortest route between each point of the upper-body and the head center. A route is a succession of continuous connections from the graph and the length of the route is equal to the sum of the used connection weights. The result can be seen on Figure 3(c): pixels with colder colours (blue, green) are those that are geodesically-closest to the top of the head; conversely, pixels with warmer colours (orange, red) are geodesically-farthest from the top of the head. The hands approximate locations are then found by computing the two biggest blobs inside the part of upper-body that are farthest from the top of the head, with typical outcome shown on Figure 3(d). Finally, as can be seen on Figure 3(e), we obtain hands locations, as well as the corresponding geodesics from head-top.

B. Features

After the tracking of the hands positions, we need to define features describing the worker's posture, see Figure 4. To do learning and recognition on several persons, we need a feature that is independent from each person's morphology.

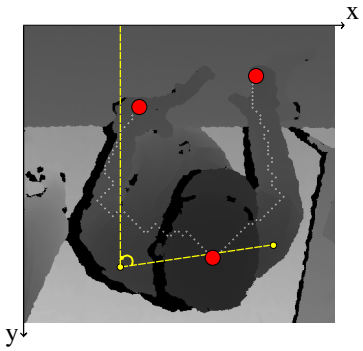

Figure 4. Representation of our features: the two hands and the head location (red dots) and the torso orientation (angle in yellow)

We chose to use the hands and head 3D locations (red dots on Figure 4) and the torso orientation (angle illustrated in yellow on 4). The third dimension of our vectors is equal to the value of the associated pixels in the depth map. These data are then concatenated in a ten-dimensions vector. For each frame, we calculate a feature which describes a posture of the worker.

C. Gestures recognition

To do the learning and the recognition we use discrete HMM, a combination of K-Means and HMM, see Figure 5. For the learning, once we have extracted the features from all the frames independently of which gesture, we use this training set to determine K clusters with the K-Means algorithm, the centroid of each cluster represents an average posture. We use this trained K-Means to quantize each series of postures, i.e., gesture. These quantized series are used as input for the HMMs for learning and recognition.

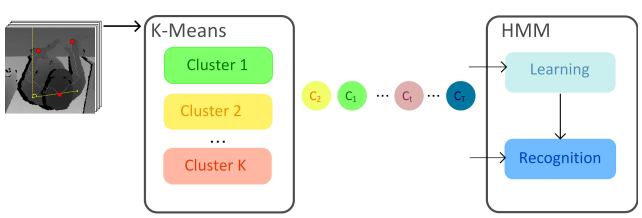

Figure 5. Pipeline of our learning and recognition method

We train one HMM for each gesture. When we want to predict which gesture is being performed, we test our quantized series of postures on each HMM. The gesture associated to the HMM with the highest likelihood to have generated this series of posture is "recognized". The Forward algorithm, described in [21], is used to establish the most likely gesture.

D. Utilisation of the inertial data

In our set-up, we also put an inertial sensor on the screwing-gun. We use this additional data source with a "late-fusion" scheme: classification by HMMs based only on features extracted from depth image is done first, and tool's inertial data is considered only afterwards to obtain the final output of our system. The screwing-gun is supposed to move only when the worker is using it to fix two parts of motor hose. We are in conflict with the result of the HMM classification in two cases:

- case 1: when the gesture G4 is recognized while the screwing gun did not move
- case 2: when a gesture which is not G4 is recognized while the screwing gun did move

For the first case, if we suppose that the inertial sensor cannot be broken, it is not possible to screw without moving the screwing-gun. Since the gesture recognized corresponds to the HMM associated with the highest likelihood to have generated the sequence of features, we look at the output likelihood of the HMM linked to gesture G4. If this likelihood is above a threshold we decide that gesture G4 has been executed, otherwise no gesture is recognized.

For the second case, it is possible that the screwing-gun moved without being used. If the worker want to move it from one side of the table to another for example. In this case we also look at the output likelihood of the HMM matching with G4. If this likelihood is above a threshold we replace the gesture previously recognized by G4, otherwise we keep the gesture associated to the HMM with the highest likelihood.

With this method, we make our system more robust by correcting wrong recognitions.

V. Results

In this Section, we will present our results of gestures recognition using either the depth-camera alone, or the combination of depth-camera with inertial sensors. We will evaluate impact of the addition of inertial sensor when we already have a depth map for our use case. We will also present a scenario of Human-Robot collaboration using our system of gesture recognition in real time. We recorded seven workers and we are using a jackknife method to evaluate our framework, i.e., we are learning HMMs with labelled gesture from six workers and testing our recognition framework on gestures of the seventh worker. We test all the seven possible combinations.

A. Using only the depth map

For our study we chose to use HMM with four hidden states and fifteen clusters for the K-Means algorithm. We obtain the results in Table I.

Table I. Gestures recognition rates with only data from the depth-camera

		Output (Maximum likelihood)					
		G1	G2	G3	G4	G5	Recall
Input Gesture	G1	**104**	15	2	3	5	81%
	G2	10	**230**	5	6	1	91%
	G3	1	7	**170**	67	10	67%
	G4	5	5	51	**180**	1	74%
	G5	2	-	2	4	**119**	94%
	Precision	85%	89%	74%	69%	88%	**80%**

We proceed at learning and recognition with all the possible combinations of six workers for the learning and one worker

for the recognition. Then, we add all the results of recognition to obtain Table I.

The recall of a gesture i represents the rate of gestures i which are recognized to be a gesture of class i. The precision is the percentage of actual i gestures among all gestures that the system labels as class i.

The total rate of correct gesture recognition is 80% which is a good result. But we can see that gesture G1 is sometimes mistaken with gesture G2. And gestures G3 and G4 are also often mistaken. Indeed the recall and precision of G3 and G4 are low: respectively 67% and 74% for G3, and 74% and 69% for G4.

B. Using combination of depth map and inertial sensors

Table II presents results of gestures recognition after correction with data from the inertial sensor fastened to the screwing-gun, as explained in Section IV-D.

Table II. Gestures recognition rates with combined data from depth-camera and inertial sensor

		Output (Maximum likelihood)					
		G1	G2	G3	G4	G5	Recall
Input Gesture	G1	**104**	15	1	-	5	83%
	G2	10	**230**	4	-	2	93%
	G3	-	3	**191**	-	9	94%
	G4	-	-	-	**242**	-	100%
	G5	1	-	4	-	**119**	96%
	Precision	90%	93%	96%	100%	88%	**94%**

We can first observe, by comparison with Table I, that all the false recognitions of G4 that were occurring without inertial sensor data, have been corrected. The recognition rate of gesture G3 is also dramatically improved. These two gestures were previously often confused one for another. With these corrections, our system reaches a global rate of correct gestures recognition of 94%, which is an excellent result, considering that our evaluation is multi-user (7 different persons repeating several times the total assembling operation). This result also highlights the interest of equipping tools with inertial sensors and use these information as a second layer after the classification of gesture with depth data.

C. Setting up a scenario of Human-Robot collaboration

We elaborate a scenario of collaboration between the worker and the robot in our use case, see Section III. To control the claws opening, the worker originally had to press two buttons, one for each claw. We removed these buttons and now use the gesture recognition to command the claw opening. To prevent a motor piece to fall on the table without a hand to grab it underneath, we combine the gesture recognition with the detection of the hand position near the claw.

For the right claw, if we recognize gesture G1 (to take a motor hose part in the right claw) and if the right hand is near the right claw, we command the claw opening. The same principle is applied for the left claw with gesture G2. This scenario enables reduction of the cognitive load of the worker because he does not have to think about the button and remains concentrated on the motor hoses assembly.

To command the claws openings we connected our computer to the robot with a Arduino [28] board. Our algorithm is coded in C++ language and processes 12 frames per second, which allows us to enable a smooth collaboration between the worker and the robot.

VI. Conclusions and futur work

In this paper, we have presented our research related to Human-Robot interaction in factories of the future. The goal is to enable tight and smooth cooperations between a human worker and a collaborative safe robot. One of the pre-requisite to attain this goal is to allow the robot to monitor and "understand" the activities of its human co-worker. We therefore focus on design, implementation and test of a method for robust and non-intrusive recognition of technical gestures, as it is clearly one of the key enabling technologies for achieving smooth Human-Robot interaction in our factory context.

We chose to use a depth-camera because it is a non-intrusive sensor robust to light changes and which provides information on the worker's posture. To augment the robustness of our system, we further equipped the tools (screwing-gun in our use-case) with inertial sensors. Our system therefore does not require the workers to wear any special equipment. We focused on a scenario of "handing" over of motor parts from a collaborative robot to a worker. Our framework differs from other studies using both a depth map and inertial sensor to do gestures recognition, firstly because in our approach inertial sensor is placed on a tool rather than on the monitored human. We first presented a new method to track hands with a top-view depth map without prior knowledge: geodesic distances from head are estimated for upper-body by Dijkstra algorithm applied on a graph of depth pixels, and hands locations are determined by searching parts that are "geodesically-farthest" from the head. For gestures recognition, we proposed a simple feature based on head and hands 3D locations and torso orientation, which is discretized by K-means and fed into discrete HMMs. We have evaluated *multi-user performances* of our system on a dataset obtained by recording 7 persons repeating several times the total assembling operation. This showed that with vision-only (the top-view depth-camera), already 80% of good gesture recognition can be attained. Furthermore, when combining with data from inertial sensor attached to the screwing-gun, the recognition rate is significantly raised to an excellent 94%. We tested our method on a simple scenario to command the robot claw opening with the gestures recognition.

Two main conclusions can be drawn from our study: 1/ robust and non-intrusive *user-independant* recognition of technical gestures of workers in factory seems feasible in principle; 2/ it can be extremely valuable for improving gestures recognition in factory to instrument workers' tools with inertial sensors. From these results, we can also conclude more generally that it should be possible to develop intelligent collaborative robots that can interact and cooperate smoothly with human workers in factories.

In our future work, we will program more reactions of the collaborative robot to the worker's gestures. We shall also record gestures from a larger pool of workers to extend our results on a larger range of morphologies and gesture executions. We will also work on an user-adaptive learning to increase our correct recognition rate with the vision only.

Copyright (c) IARIA, 2016. ISBN: 978-1-61208-468-8

ACKNOWLEDGMENT

This research benefited from the support of the Chair 'PSA Peugeot Citroën Robotics and Virtual Reality', led by MINES ParisTech and supported by PEUGEOT S.A. The partners of the Chair cannot be held accountable for the content of this paper, which engages the authors´ responsibility only.

REFERENCES

[1] A. Bauer, D. Wollherr, and M. Buss, "Human–robot collaboration: a survey," International Journal of Humanoid Robotics, vol. 5, 2008, pp. 47–66.

[2] M. L. Walters, K. L. Koay, D. S. Syrdal, A. Campbell, and K. Dautenhahn, "Companion robots for elderly people: Using theatre to investigate potential users' views," Proceedings - IEEE International Workshop on Robot and Human Interactive Communication, 2013, pp. 691–696.

[3] S. Thrun and M. Bennewitz, "MINERVA: A second-generation museum tour-guide robot," Proceedings of IEEE International Conference on Robotics and Automation, vol. 3, 1999.

[4] T. Sakaki, N. Ushimi, K. Aoki, K. Fujiie, R. Katamoto, A. Sugyo, and Y. Kihara, "Rehabilitation robots assisting in walking training for SCI patient," Proceedings - IEEE International Workshop on Robot and Human Interactive Communication, 2013, pp. 814–819.

[5] E. Sato, T. Yamaguchi, and F. Harashima, "Natural Interface Using Pointing Behavior for Human–Robot Gestural Interaction," IEEE Transactions on Industrial Electronics, vol. 54, 2007, pp. 1105–1112.

[6] K. Nickel and R. Stiefelhagen, "Visual recognition of pointing gestures for human–robot interaction," Image and Vision Computing, vol. 25, 2007, pp. 1875–1884.

[7] "Kuka Robotics," URL: http://www.kuka-robotics.com/fr [accessed: 2016-03-18].

[8] "Universal Robot UR," URL: http://www.universal-robots.com/fr/produits/robot-ur5/ [accessed: 2016-03-18].

[9] T. Wojtara, M. Uchihara, H. Murayama, S. Shimoda, S. Sakai, H. Fujimoto, and H. Kimura, "Human-robot collaboration in precise positioning of a three-dimensional object," Automatica, vol. 45, no. 2, 2009, pp. 333–342.

[10] V. Weistroffer, A. Paljic, P. Fuchs, O. Hugues, J.-P. Chodacki, P. Ligot, and A. Morais, "Assessing the acceptability of human-robot co-presence on assembly lines: A comparison between actual situations and their virtual reality counterparts," in The 23rd IEEE International Symposium on Robot and Human Interactive Communication. IEEE, aug 2014, pp. 377–384.

[11] S. Nikolaidis, R. Ramakrishnan, K. Gu, and J. Shah, "Efficient Model Learning from Joint-Action Demonstrations for Human-Robot Collaborative Tasks," in Proceedings of the Tenth Annual ACM/IEEE International Conference on Human-Robot Interaction, ser. HRI '15. ACM, 2015, pp. 189–196.

[12] "MicrosoftKinect," URL: https://dev.windows.com/en-us/kinect [accessed: 2016-03-18].

[13] J. Han, L. Shao, D. Xu, and J. Shotton, "Enhanced Computer Vision with Microsoft Kinect Sensor: A Review," IEEE Trans. Cybernetics, vol. 43, no. 5, 2013, pp. 1318–1334.

[14] L. A. Schwarz, A. Mkhitaryan, D. Mateus, and N. Navab, "Human skeleton tracking from depth data using geodesic distances and optical flow," Image and Vision Computing, vol. 30, no. 3, 2012, pp. 217–226.

[15] J. Shotton, A. Fitzgibbon, M. Cook, T. Sharp, M. Finocchio, R. Moore, A. Kipman, and A. Blake, "Real-time Human Pose Recognition in Parts from Single Depth Images," in Proceedings of the 2011 IEEE Conference on Computer Vision and Pattern Recognition, ser. CVPR '11. IEEE Computer Society, 2011, pp. 1297–1304.

[16] C. Migniot and F.-E. Ababsa, 3D Human Tracking in a Top View Using Depth Information Recorded by the Xtion Pro-Live Camera, ser. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2013, vol. 8034.

[17] C. Jost, P. De Loor, L. Nedelec, E. Bevacqua, and I. Stankovic, "Intelligent Technologies for Interactive Entertainment (INTETAIN), 2015 7th International Conference on," pp. 47–56, 2015.

[18] L. Miranda, T. Vieira, D. Martinez, T. Lewiner, A. W. Vieira, and M. F. M. Campos, "Real-Time Gesture Recognition from Depth Data through Key Poses Learning and Decision Forests." IEEE, aug 2012, pp. 268–275.

[19] K. Lai, J. Konrad, and P. Ishwar, "A gesture-driven computer interface using Kinect," 2012 IEEE Southwest Symposium on Image Analysis and Interpretation, 2012, pp. 185–188.

[20] A. Ramey, V. González-Pacheco, and M. A. Salichs, "Integration of a low-cost RGB-D sensor in a social robot for gesture recognition," in Proceedings of the 6th international conference on Human-robot interaction - HRI '11. New York, New York, USA: ACM Press, 2011, p. 229.

[21] L. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," Proceedings of the IEEE, vol. 77, no. 2, 1989, pp. 257–286.

[22] B. Lovell, P. Kootsookos, and R. Davis, "Model Structure Selection and Training Algorithms for an HMM Gesture Recognition System," Ninth International Workshop on Frontiers in Handwriting Recognition, 2004, pp. 100–105.

[23] V. Kellokumpu, M. Pietikäinen, and J. Heikkilä, "Human activity recognition using sequences of postures." MVA, 2005, pp. 570–573.

[24] J. Hol, T. Schon, F. Gustafsson, and P. Slycke, "Sensor fusion for augmented reality," in 2006 9th International Conference on Information Fusion. IEEE, 2006, pp. 1–6.

[25] K. Liu, C. Chen, R. Jafari, and N. Kehtarnavaz, "Fusion of Inertial and Depth Sensor Data for Robust Hand Gesture Recognition," IEEE Sensors Journal, vol. 14, no. 6, 2014, pp. 1898–1903.

[26] E. Coupeté, S. Manitsaris, and F. Moutarde, "Real-time recognition of human gestures for collaborative robots on assembly-line," in 3rd International Digital Human Modeling Symposium (DHM2014), 2014, pp. 7–p.

[27] E. W. Dijkstra, "A note on two problems in connexion with graphs," Numerische Mathematik, vol. 1, no. 1, 1959, pp. 269–271, ISSN:0029599X.

[28] "Arduino," URL: https://www.arduino.cc/ [accessed: 2016-03-18].