A User-Adaptive Gesture Recognition System Applied to Human-Robot Collaboration in Factories

Eva Coupeté, Fabien Moutarde, Sotiris Manitsaris

A User-Adaptive Gesture Recognition System Applied to Human-Robot Collaboration in Factories

Eva Coupeté
Center of Robotics
Mines ParisTech
PSL research University
60 Bd Saint Michel 75006
Paris, France
eva.coupete@mines-paristch.fr

Fabien Moutarde
Center of Robotics
Mines ParisTech
PSL research University
60 Bd Saint Michel 75006
Paris, France
fabien.moutarde@mines-paristech.fr

Sotiris Manitsaris
Center of Robotics
Mines ParisTech
PSL research University
60 Bd Saint Michel 75006
Paris, France
sotiris.manitsaris@mines-paristech.fr

ABSTRACT

Enabling Human-Robot collaboration (HRC) requires robot with the capacity to understand its environment and actions performed by persons interacting with it. In this paper we are dealing with industrial collaborative robots on assembly line in automotive factories. These robots have to work with operators on common tasks. We are working on technical gestures recognition to allow robot to understand which task is being executed by the operator, in order to synchronize its actions. We are using a depth-camera with a top view and we track hands positions of the worker. We use discrete HMMs to learn and recognize technical gestures. We are also interested in a system of gestures recognition which can adapt itself to the operator. Indeed, a same technical gesture seems very similar from an operator to another, but each operator has his/her own way to perform it. In this paper, we study an adaptation of the recognition system by modifying the learning database with a addition very small amount of gestures. Our research shows that by adding 2 sets of gestures to be recognized from the operator who is working with the robot, which represents less than 1% of the database, we can improve correct recognitions rate by $\sim$3.5%. When we add 10 sets of gestures, 2.6% of the database, the improvement reaches 5.7%.

Keywords

Human-robot collaboration; Gesture recognition; User adaptation; Assembly line; Depth camera.

Categories and Subject Descriptors

I.2.9. [**Robotics**]: Operator interfaces
I.2.10. Vision and Scene Understanding: Miscellaneous
H.5.2. User Interface: Interaction styles

MOCO'16, July 05-06, 2016, Thessaloniki, GA, Greece

DOI: **http://dx.doi.org/10.1145/2948910.2948933**

1. INTRODUCTION

Collaborative robots tend to be more and more present in factories. Indeed, they allow more automation on supply chains, which saves place and cost while increasing productivity. These robots can work with operators in a common workstation on low added-value tasks or tasks source of musculoskeletal disorders. To enable a smooth collaboration, the robot has to be synchronized with the operator, and therefore needs to understand which action has been executed by the operator. To allow the robot to perceive its environment is also necessary for ensuring a safe collaboration.

Gesture recognition can be a solution to facilitate a fluid and safe collaboration. By recognizing worker's gesture, the robot can understand which task has been executed, can adapt his speed, and can react properly if something unexpected happens. We chose to use a depth camera to be able to detect a situation of danger, when a robot and an operator are too close for example. One of the difficulties of this goal is to have a gesture recognition system that can correctly recognize gestures made by a large number of operators, without disturbing them during their work. Indeed, when an operator is working on a supply chain next to a collaborative robot, he will not repeat an action to be certain that the robot understand correctly what he just did. We chose to use HMMs on our gesture recognition pipeline. Each HMM represents a technical gesture and it is learnt with a large number of examples of this gesture executed by several operators. But each operator has his/her own way to perform a gesture which could lead to mistake in recognition. We want to know if an adaptation of the gestures recognition pipeline to a new operator, which is not in the leaning database, could improve our correct recognition rate. We imagine that if we could record some gestures from this new operator we could just modify the learning database with this few number of gestures. Also, we do not work on adaptive HMMs, we chose to learn again the HMMs with this new database. Indeed, in our use case, the time to learn the HMMs is below 5 minutes, which is a quite reasonable "loss of time" if we project to record a new operator with the aim to make his/her collaboration with a robot more productive and robust during several days of work.

This paper is organized in five parts. In the first part we will present related work, in the second part we will introduce our use case and in the third part we will explain our pipeline of gesture recognition from the depth map images to recognized gesturse. In the fourth part we will show our methods of evaluation and in the last part we will conclude and discuss about this study.

2. RELATED WORK

In this section we present related work on the topics of human-robot collaboration and gesture recognition.

2.1 Human Robot collaboration

With the automation of factories and the arrival of interactive robots in our everyday life, research on human-robot collaboration has been very active these last years [4]. Robots are already used to help children with autism [16], to interact with elderly people [21] or to guide visitors in a museum [20]. For these applications, the robots used are mainly anthropomorphic. For the industry, collaborative robots are designed to be safe and to provide complementary skill to human co-worker like the Kuka LWR [2] and the Universal Robot UR [3]. They work nearby the operator, on a common task, like carrying the same object [24]. In [18] the authors categorize robotic systems in low, medium and high levels of human-robot collaboration. In [22] the authors evaluate the worker's acceptability to work with a collaborative robot. In [14] the authors showed that working with a robot adds uncertainty about the worker's next action. These last studies illustrate the high interest and potential of the insertion of collaborative robot in factories, but also all existing difficulties to create a productive and safe collaboration.

2.2 Gesture recognition

Since the technical breakthrough in human-computer interaction, gesture recognition has been a well studied field. The emergence of depth-cameras allowed to make improvement in human body skeleton tracking. In [19] the authors used randomized decision forests trained with a large number of training depth images and a simple depth comparison features to find the joint position of a human filmed with a depth camera. In [17], the authors do not have a learning database to train a system to find the joints positions, therefore they use geodesic distances of a human body to track them. Numerous methods to recognize dynamic gestures have been set up. The most known are HMM (Hidden Markov Model) [15], [12] and [10]. Other machinel-learning alogorithm are used, like SVM (Support Vector Machine); Dynamic Time Wrapping (DTW) [8], decision forest or K-Nearest Neighbours.

The adaptation of HMMs to user has, in a first place, been studied for speaker adaptation. In [11] a speaker-independent system is adapted to a new speaker by updating the HMMs parameters. More recently, user adaptation has been applied on video based studies, like for face recognition in [25] or gesture recognition [23]. Adaptation process has also been applied to DTW for gesture recognition by updating the estimated parameters to provide a continuous human-machine interaction [5]. More precisely in the human-robot collaboration field, the authors of [13] adapt their learned models to different human types after a first joint-action demonstrations step with the new user and the robot. They have shown that this adaptation led to an improvement in team fluency and a decrease in the human idle time.

3. PRESENTATION OF THE 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, 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 according 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 two 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:

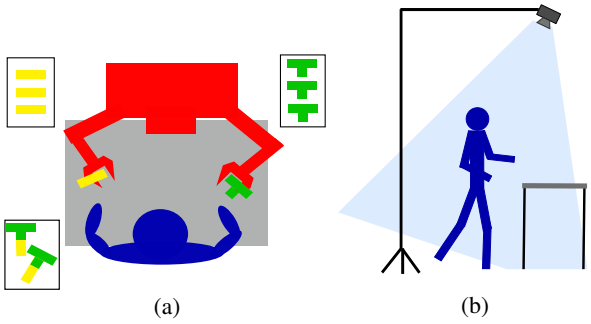

Figure 1: Description of our experimental use-case, (a): the robot gives motor parts to the worker, (b): we equipped the scene with a depth-camera

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, see Figure 1(b).

4. METHODOLOGY

In this section we explain our methodology to achieve technical gesture recognition. Using segmentation and computation of geodesics on top-view depth image, we estimate a six-dimensions feature vector used for gesture recognition, see the first subsection. Geodesics represents the distance between the top of the head and every pixel on the torso, following the body shape. In the second subsection we present our pipeline for gestures learning and recognition using a hybrid system of K-Means and HMM.

4.1 Features extraction

To localize and track hands of the worker, we have adapted and extended to top-view case the method proposed in [17].

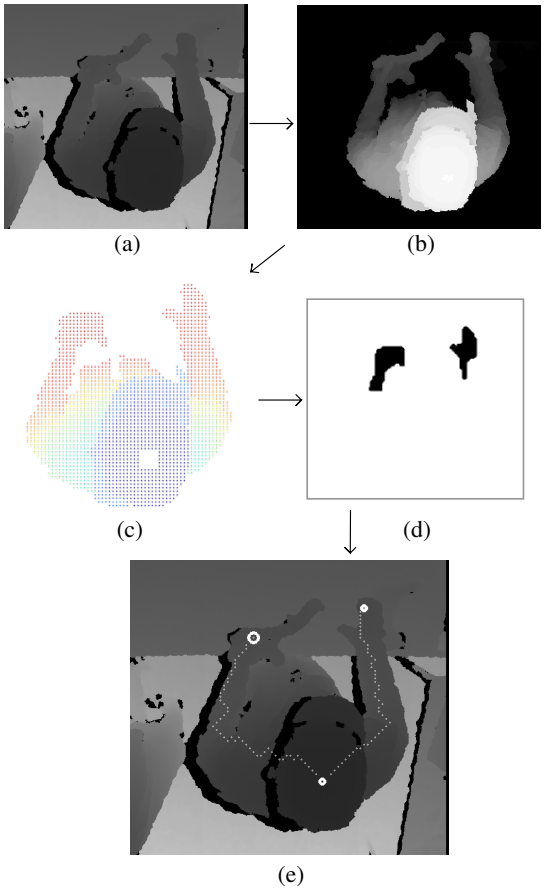

Figure 2: Our hands-tracking method. (a): initial depth map, (b): upper body of the worker, (c): geodesic distances between each point of the upper body and the head (blue to red), (d): farthest upper body parts from the head, (e): head and hands locations with the shortest paths between the hands and the head

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 2(a)), we segment the upperbody by keeping only depth pixels that are above the assembling table (see typical result on Figure 2(b)). The top of the head is located as the pixels of the upper body nearest to the depth camera. In order to locate hands, we make the assumption that they are the visible parts of the upper-body that are farthest from the head, not in Euclidean straight line distance, but following the body surface. To find these "farthest" points, we calculate geodesic distances between head-top and all points of the upper-body. We apply Dijkstra [7] algorithm in order to find the shortest route between each point of the upper-body and the head center. The result can be seen on Figure 2(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 2(d). Finally, as can be seen on Figure 2(e), we obtain hands locations, as well as the corresponding geodesics from head-top.

After the tracking of the hands positions, we need to define features describing the worker's posture, see Figure 3. To enable learning and recognition on several persons, we need a feature that is independent from each person's morphology.

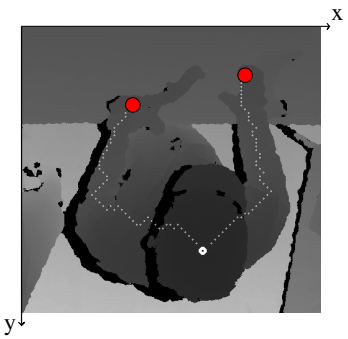

Figure 3: Representation of our features : the two hands (red dots)

We chose to use the hands 3D locations (red dots on Figure 3). 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 six-dimensions vector.

4.2 Pipeline of gesture recognition

To do learning and recognition we use discrete HMM, a combination of K-Means and HMM, see Figure 4.

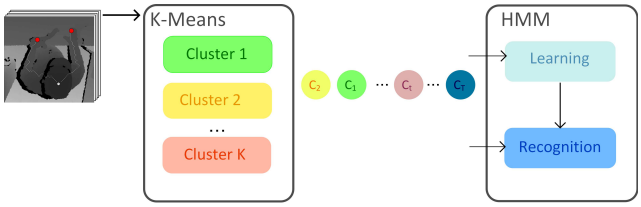

Figure 4: Pipeline of our learning and recognition method

For learning, once we have extracted 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. We train one HMM for each gesture. When we want to estimate 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 [15], is used to establish the most likely gesture.

5. RESULTS

In this section we present the performances of our gesture recognition pipeline. In the two first subsections we introduce two methods to evaluate our pipeline and show the results. In the two subsections following we present our user adaptation database and the

results of correct recognition. Finally, in the last subsection, we explain how we implemented our gesture recognition pipeline.

5.1 Methods of evaluation of our gesture recognition pipeline

To evaluate the performance of the gesture recognition pipeline we use two methods.

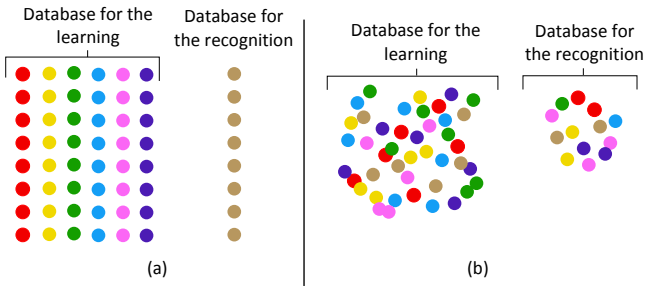

Figure 5: Illustration of our gestures database division according to which method of evaluation we use. Each color is associated to an operator in the database, each dot is a gesture. (a) the jackknife method, (b) the 80%-20% method.

The first method, called jackknife, allows assessment of the recognition of gestures performed by an operator of whom no gestures are in the learning database. In practice, we learn the HMMs with gestures from all operators from our database except one, and we test recognition on this last operator. We evaluate our system by testing all possible combinations of (N-1 operators) for learning, 1 operator for recognition. For N operators, we have N combinations. This method allow us to evaluate the recognition rates for new operator working with the robot.

The second method, called 80%-20%, evaluates the recognition system when the database for learning is composed of 80% of all gestures of the entire database. The 20% left are used to evaluate the result therefore estimates the gestures recognition rate for operators included in training database. Contrary to the jackknife, with the 80%-20% method, we evaluate the recognition rate for operators who already are in the database.

5.2 Evaluation of our gesture recognition pipeline

We have a database of 13 operators for this study. We recorded the operators working with the robot, doing the assembly task between 20 to 25 times each. The results presented below were computed with a K-Means of 20 clusters and a HMMs with 7 states.

5.2.1 *The jackknife method*

The Table 1 include all the results for all the combinations of (N-1) operators for the learning and 1 operator for the recognition.

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

We obtain a good result of 80% of correct recognitions. We can observe some mistake between gestures G1 and G2 and between gestures G3 and G4. Indeed the recall of G4 is 72% and almost 20% of gestures G4 are recognized as gesture G3. These two gestures, to join two parts of the motors hose (G3) and to screw (G4), look very similar from a top view with a depth camera. For both, the operator is holding pieces and have the hands almost clasped in front of him. The ambiguity between gestures G1, to take a motor hose part in the robot right claw, and G2, to take a motor hose part in the robot left claw, can be explained by the fact that the operator are sometimes doing these two actions simultaneously, at the beginning of a cycle, when the operator has to take a piece in each claw.

Table 1: Gestures recognition rates for unknown operators, estimated by jackknife method

		Output (Maximum likelihood)					
		G1	G2	G3	G4	G5	Recall
Input Gesture	G1	**141**	19	1	6	1	84%
	G2	38	**232**	-	22	-	79%
	G3	21	0	**214**	32	10	77%
	G4	15	10	56	**193**	1	72%
	G5	1	-	2	-	**146**	98%
	Precision	65%	89%	78%	76%	92%	**80%**

5.2.2 *The 80%-20% method*

We have the same database with 13 operators, as for the jackknife method. We used 80% of gesture from our database to learn the HMMs and we used the 20% left to evaluate recognition. Results, given in Table 2, provide an estimation of recognition performances for operators included in training database.

Table 2: Gestures recognition rates for operators included in training set, estimated by 80%-20% method

		Output (Maximum likelihood)					
		G1	G2	G3	G4	G5	Recall
Input Gesture	G1	**46**	11	2	2	-	75%
	G2	7	**103**	-	8	-	87%
	G3	-	-	**88**	8	2	90%
	G4	2	2	10	**62**	2	79%
	G5	0	2	1	1	**51**	93%
	Precision	84%	87%	87%	77%	93%	**85%**

The average correct recognition rate obtained is 85%, which is 5% higher than or previously unseen operators (as estimated by jackknife).

5.2.3 *Comparison with DTW*

We compared our pipeline with a method using DTW. The features are the same that above, but they are not discretized, and are directly used to train the DTW templates and to recognize gestures. We used the method explained in [9] called ND-DTW. This method allows using DTW with N-dimensional features. Each gesture template corresponds to the gesture in the training base that minimizes the normalized wrapping distance to the other gestures in the training set. We have the same dataset that we used for the HMM pipeline.

With the jackknife method we obtain a very low accuracy of gesture recognition of 20%. This result is improved using the 80%-20% method by 3%, but remains low, 23% of correct recognition. This can be explained by the computation of the gesture template. Using one gesture on the database to be a template is not robust enough to the gesture variations. Indeed, we have gestures of 13 operators who all have their own way to perform the technical gestures we want to recognize. A probabilistic method, like HMM, is

a more suitable solution to describe these variations. In the rest of the study, we will then concentrate on HMM.

5.3 User-adaptive gesture recognition system by modifying the learning database

As presented above, we observe that gestures recognition rate is 4% higher for users included in training dataset (cf. 80%-20% evaluation compared to jackknife evaluation). This is rather natural and somewhat expected. An interesting issue is therefore to define a methodology for user-adaptive recognition. A simple way to handle new unknown operator is to enrich training database with at least a few examples of his way of performing gestures, and then retrain. We therefore conducted experiments to evaluate what would be the minimal number of recordings of the new operator to be added, in order to obtain for him performances similar to those for operators included in the initial large training dataset. For this purpose, we modified the jackknife method by adding a small amount of the "test" operator gestures in the learning database and test the system on the remaining gestures from this same "test" operator, see Figure 6.

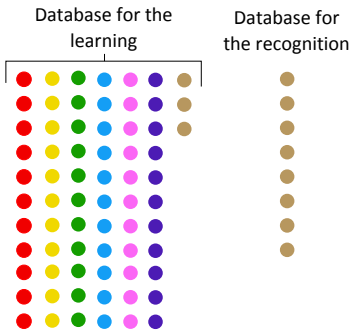

Figure 6: Illustration of our method to modify the learning database. Each color is associated to an operator in the database, each dot is a gesture.

We evaluate the result of recognition with different amount of gestures from the "test" operator in the learning database. These new training examples gesture are added by set of the five gestures we want to recognize. One set is equal to one example of each of these five gestures, two sets is two examples, etc. We compare the results obtained with the new learning database with those obtained before using standard jackknife method for the same "test" operator. The results presented below are an average of the results on 6 combinations of (N-1) operators for the learning, 1 for the testing, with N still equal to 13. In future work, we will extend these results with all possible combinations.

5.4 Evaluation of our user adaptive gesture recognition system

We injected only small numbers of gestures from the "test" operator in the learning database for several reasons. The first one is that if we ask an operator to perform some gestures to adapt the system to his own way to work, it is likely that this recording will be short and only a small number of gestures will be added in the learning database. The second reason is that if we inject a large number of gestures form the "test" operator to the learning database, we will not have enough remaining examples to evaluate properly the recognition system.

We can see on Table 3 that with a small number of gestures, 2 set or 10 gestures, from the test operator injected in the learning database, the correct recognition rate can quickly improve improve by 3.5%. This implies that recording only 2 cycles of a new operator and adding those gestures examples to training dataset, seems enough to recover recognition performance similar to that for operators included in initial database. In our case, 10 gestures represent less than 0.5% of the gestures in the learning database. These results allow us to expect that with a small number of gestures from an operator, it can be easy and fast to adapt the gesture recognition system to his own way of making gestures and improve the performance of the system.

Table 3: Improvement of the gesture recognition rates

	Number of sets added					
	1	2	3	4	5	10
Improvement of correct recognition rate	2.4%	3.5%	3.6%	3.1%	4.1%	5.7%

We also looked at the impact of the new gestures in the learning database on the precision and recall rates. We can see the results for the evolution of the precision in Table 4 and of the recall in Table 5. The numbers are in green when the precision or recall improved by more than 2.5%. They are in red if they reduce more than 1%.

Table 4: Precision evolution in percent

		Gesture class				
		G1	G2	G3	G4	G5
Number of set added	**1**	-1.9	-1.1	2.9	6.1	0.0
	2	0.1	0.1	0.5	7.3	-1.1
	3	-0.8	0.4	3.6	5.9	0.2
	4	-2.2	1.0	3.2	5.7	-0.8
	5	0.1	0.5	3.6	8.9	0.2
	10	6.3	3.7	3.7	16.4	-0.9

Table 5: Recall evolution in percent

		Gesture class				
		G1	G2	G3	G4	G5
Number of sets added	**1**	-0.4	-0.1	0.6	6.9	-0.4
	2	-3.2	0.5	2.9	8.8	-0.4
	3	-0.3	1.0	1.9	8.8	0.0
	4	1.1	0.3	3.0	6.1	-2.2
	5	-0.1	0.6	1.9	7.1	-2.3
	10	-0.8	3.1	5.4	8.1	6.0

In both cases, precision and recall, we can observe a significant improvement for gesture G4, to screw. This gesture was sometimes mistaken with gesture G3, to join two parts of the motor hose. There is also improvements in the precision and recognition rates for gesture G3. It seems there are less mistakes between these two gestures when we add gestures from the "test" operator in the learning database.

For gesture G5, to put the final motor hose in a box, there is not a great improvement, the rates stay quite similar to the result with the jackknife. Indeed, gestures G5 was already well discerned from the other gestures and had good rates of recall and precision.

Similarly for gestures G1, to take a motor hose part in the robot right claw, and G2, to take a motor hose part in the robot left claw,

there is no noticeable improvement. The rates tend to stay similar as they were with the jackknife.

It seems that to adapt the database to a system user can lead to an improvement for gestures with high confusion, but do not improve the recognition of gestures which are already well discriminated from others.

5.5 Implementation and computation time

We implemented our gesture recognition pipeline on C++ using the GRT library [1].

In Table 6 we present the computation time of our learning system. The numbers in green correspond to a computation time below 1 minute, in orange between 1 and 2 minutes 30 seconds and red above 2 minutes 30 seconds.

Table 6: Computation time for the learning, in seconds

		Number of states				
		4	5	7	10	15
Number of clusters	**10**	26	37	59	101	219
	15	32	43	70	124	284
	20	39	52	79	138	325
	25	45	61	95	161	351

Even if HMM training time increases with the parameters size, it stays quite short. It is conceivable to easily remake a learning of the HMMs when a new operator is working on a collaborative task with a robot without a large lack of time.

The time to recognize a gesture, for all the configurations presented on Table 6, is below the millisecond.

We looked at the evolution of the computation time according to the database size, see Figure 7. We plotted on the same graph the computation time for different database sizes and a linear regression of these data. The database used for this graph is the same that we used for the study. Each dot on the Figure 7 represents the training time of a K-Means of 10 clusters and 5 HMMs with 7 states.

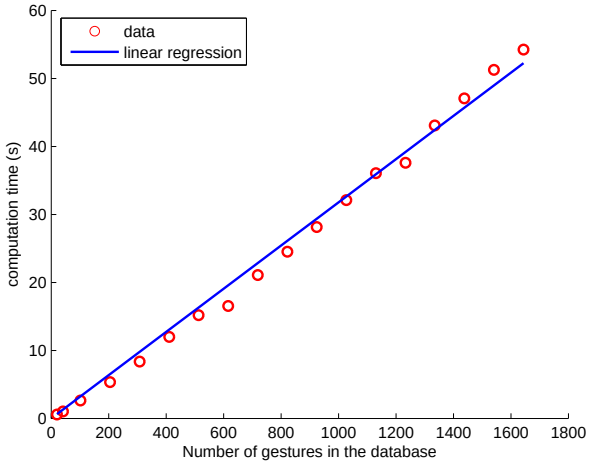

Figure 7: Computation time for the learning depending on the database size; red dots : the data computed using our pipeline, blue line : the linear regression

We can observe that the computation time seems to linearly evolve with the database size. Indeed, the learning is done, for each sequence in the learning base, by running the Forward-Backward algorithm and then using the Baum-Welch algorithm to reestimate the HMM parameters, [15]. The complexity of the learning algorithm is then linearly dependent of the number of example in the learning base. The slope of the linear regression we can see Figure 7 is equal to 0.0318, which means that, in average, for each new gesture exemple in the leaning base, the learning time will increase by 0.0318s. If we add 10 sets of gestures in learning base, which represents 50 gestures, the learning time will increase in average by 1.59s, which is negligible.

To add new gesture to our sytem, no new calibaration is needed. We just have to record a fixed number of gestures, 25 for 5 sets of new gestures. On our previous recordings, we observed that the operators are peforming 20 gestures in 3 min approximately. Then, a segmentation and labelling step is needed. It is the longest part of the process, but for this amount of gestures (25 for 5 sets), it takes less than an hour. Then the learning step is quite short, as we can see above. We could improve the process by automating the segmentation and labelling step by doing separate recordings for each gesture, which will save time.

6. CONCLUSION

In this paper we have presented our research related to human-robot collaboration in factories. The goal is to enable a smooth and robust collaboration between a human worker and a collaborative safe robot. To this end, it is necessary to provide to the robot a capacity to understand what the worker is doing. This is why we studied gesture recognition with a depth camera to have information on the worker posture and his distance to the robot. We track hands using geodesic distances and the Dijkstra algorithm. For the gesture recognition we use the 3D locations of the two hands as features. We discretized all possible locations using K-Means and use the output clusters to train our HMM.

We have collected a large database with 13 different operators, each one recorded while performing 20 assembling cycles (providing a total of 160 gestures per operator). Using this database, we obtain 85% correct gesture recognition for operators included in training set. We also evaluated the expected recognition rate for "new" operators (i.e. not present in training data) as 80%. Motivated by this significant performance difference, we studied the possibility to adapt the gesture recognition pipeline to new worker in order to improve its correct recognitions rates. We supposed that it could be possible to record a few examples of gestures from the new operator to modify the learning database of the HMM, and then learn them again (which takes at most $\sim$6 minutes of computation time). We observed that even a small amount of added gestures can significantly improve the correct gesture recognition rates: more than 3,5% with only 2 sets of gestures, which represents 1,5% of the learning database.

This method could be applied to numerous gestural interfaces. The varibility in the gestures performed by severel people can also be an issue to enable an natural communication between a person and a digital interface. Also, recording new gestures from a new user to modify the leanring database seems to be a quick and robust way to improve the accuracy of the gesture recognition.

In order to further increase recognition performances in our use case, we work in parallel on the addition of inertial sensors on tools, the screwing-gun for this use case, to have new information on the worker's actions, [6]. The advantage of the inertial sensors on tools, rather than on the operator himself, is that he does not have to wear any equipment. Combined with the user-adaptive method-

ology presented and evaluated here, it should be possible to obtain a very robust system allowing a safe, reliable and efficient collaboration between operator and robot.

7. REFERENCES

[1] Gesture Recognition Toolkit http://www.nickgillian.com/software/grt.

[2] KUKA Robotics http://www.kuka-robotics.com/fr.

[3] Universal Robots http://www.universal-robots.com/fr/produits/robot-ur5/.

[4] BAUER, A., WOLLHERR, D., AND BUSS, M. Human–robot collaboration: a survey. *International Journal of Humanoid Robotics 5*, 01 (2008), 47–66.

[5] CARAMIAUX, B., MONTECCHIO, N., TANAKA, A., AND BEVILACQUA, F. Adaptive Gesture Recognition with Variation Estimation for Interactive Systems. *ACM Transactions on Interactive Intelligent Systems 4*, 4 (dec 2014), 1–34.

[6] COUPETÉ, E., MOUTARDE, F., MANITSARIS, S., AND HUGUES, O. Recognition of Technical Gestures for Human-Robot Collaboration in Factories. In *The Ninth International Conference on Advances in Computer-Human Interactions* (2016).

[7] DIJKSTRA, E. W. A note on two problems in connexion with graphs. *Numerische Mathematik 1*, 1 (1959), 269–271.

[8] GAVRILA, D. M., AND DAVIS, L. S. Towards 3-D model-based tracking and recognition of human movement: a multi-view approach. *In International Workshop on Automatic Face- and Gesture-Recognition. IEEE Computer Society* (1995), 272–277.

[9] GILLIAN, N., KNAPP, R. B., AND O 'MODHRAIN, S. Recognition Of Multivariate Temporal Musical Gestures Using N-Dimensional Dynamic Time Warping. *NIME* (2011), 337–342.

[10] KELLOKUMPU, V., PIETIKÄINEN, M., AND HEIKKILÄ, J. Human activity recognition using sequences of postures. *MVA* (2005), 570–573.

[11] LEGGETTER, C., AND WOODLAND, P. Maximum likelihood linear regression for speaker adaptation of continuous density hidden Markov models. *Computer Speech & Language 9*, 2 (apr 1995), 171–185.

[12] LOVELL, B., KOOTSOOKOS, P., AND DAVIS, R. Model Structure Selection and Training Algorithms for an HMM Gesture Recognition System. *Ninth International Workshop on Frontiers in Handwriting Recognition* (2004), 100–105.

[13] NIKOLAIDIS, S., RAMAKRISHNAN, R., GU, K., AND SHAH, J. 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* (New York, NY, USA, 2015), HRI '15, ACM, pp. 189–196.

[14] NIKOLAIDIS, S., AND SHAH, J. Human-robot cross-training: computational formulation, modeling and evaluation of a human team training strategy. In *Proceedings of the 8th ACM/IEEE International Conference on Human-robot Interaction* (mar 2013), IEEE Press, pp. 33–40.

[15] RABINER, L. A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE 77*, 2 (1989), 257–286.

[16] ROBINS, B., DICKERSON, P., STRIBLING, P., AND DAUTENHAHN, K. Robot-mediated joint attention in children with autism: A case study in robot-human interaction. *Interaction Studies 5*, 2 (2004), 161–198.

[17] SCHWARZ, L. A., MKHITARYAN, A., MATEUS, D., AND NAVAB, N. Human skeleton tracking from depth data using geodesic distances and optical flow. *Image and Vision Computing 30*, 3 (mar 2012), 217–226.

[18] SHI, J., JIMMERSON, G., PEARSON, T., AND MENASSA, R. Levels of human and robot collaboration for automotive manufacturing. In *Proceedings of the Workshop on Performance Metrics for Intelligent Systems - PerMIS '12* (New York, New York, USA, mar 2012), ACM Press, p. 95.

[19] SHOTTON, J., FITZGIBBON, A., COOK, M., SHARP, T., FINOCCHIO, M., MOORE, R., KIPMAN, A., AND BLAKE, A. Real-time Human Pose Recognition in Parts from Single Depth Images. In *Proceedings of the 2011 IEEE Conference on Computer Vision and Pattern Recognition* (Washington, DC, USA, 2011), CVPR '11, IEEE Computer Society, pp. 1297–1304.

[20] THRUN, S., AND BENNEWITZ, M. MINERVA: A second-generation museum tour-guide robot. *Proceedings of IEEE International Conference on Robotics and Automation 3* (1999).

[21] WALTERS, M. L., KOAY, K. L., SYRDAL, D. S., CAMPBELL, A., AND DAUTENHAHN, K. Companion robots for elderly people: Using theatre to investigate potential users' views. *Proceedings - IEEE International Workshop on Robot and Human Interactive Communication* (2013), 691–696.

[22] WEISTROFFER, V., PALJIC, A., FUCHS, P., HUGUES, O., CHODACKI, J.-P., LIGOT, P., AND MORAIS, A. 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* (aug 2014), IEEE, pp. 377–384.

[23] WILSON, A., AND BOBICK, A. Realtime online adaptive gesture recognition. In *Proceedings 15th International Conference on Pattern Recognition. ICPR-2000* (2000), vol. 1, IEEE Comput. Soc, pp. 270–275.

[24] WOJTARA, T., UCHIHARA, M., MURAYAMA, H., SHIMODA, S., SAKAI, S., FUJIMOTO, H., AND KIMURA, H. Human-robot collaboration in precise positioning of a three-dimensional object. *Automatica 45*, 2 (2009), 333–342.

[25] XIAOMING LIU, AND TSUHAN CHENG. Video-based face recognition using adaptive hidden Markov models. In *2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2003. Proceedings.* (2003), vol. 1, IEEE Comput. Soc, pp. I–340–I–345.