Learning Operator Intentions Using Supervised Learning for Safe Human-Robot Collaboration

Iker Ceballos

Abstract—**Human behaviour prediction is important to enable proper and safe collaboration between humans and robots working within the same workspace. In an industrial environment where not all the tasks can be automated, a human worker and a robot can be performing tasks simultaneously in a common workspace e.g. picking pieces from the same conveyor belt. In this case the conventional approach of stopping the robot whenever it is going to collide with the worker is not an optimal solution as it decreases the actual cycle time of the robot and therefore the productivity. In this paper a way of learning the intentions of the human worker, so that the robot can still operate safely, is sought based on 3D-sensor feedback, a skeleton model of the human and a supervised classification algorithm. *Partitioning Around Medoids* (*PAM*) is used to separate the recorded tracks into classes and then *SVM* is trained with these data. Subsequently, real-time recording is classified with the trained model to allow early prediction of the action the operator is performing.**

Index Terms—**collaborative robots, *SVM*, *PAM*, human intention anticipation.**

I. Introduction

The usage of robots has revolutionized manufacturing. However, there are still tasks that require human dexterity and cannot be fully automated, thus collaborative robots are said to be one of the upcoming revolutions in industry [1]. Collaborative robots are able to work hand by hand with human workers, but, according to *ISO 10218-1*, for safety reasons they need to incorporate safety-rated monitored speed and power functions which conventional industrial robots which are caged and working on their own don't need. Therefore, it would be desirable for a collaborative robot to be able to work as close as possible to full capacity while safeguarding the human at the same time.

In this paper unsupervised clustering techniques such as *PAM* are used to automatically separate the actions performed within the workspace into classes, the centroids of the generated clusters can be considered as the prototypical way of performing each action. It then combines these results with a supervised learning model such as *SVM* to classify new observed behaviours into the learned classes in real-time. Based on these techniques this work presents a novel way to classify human action to allow early prediction and provides a novel strategy to make these predictions.

Specifically, this paper contributes presenting a novel way to define and identify different human actions within a known workspace so that they are easily distinguishable for a classifier. It also presents a novel strategy to detect the beginning of and early predicting a human action.

Fig. 1. Skeletal tracking during a manual assembly process, a robot working in the same workspace as the operator can gain knowledge of where every joint composing this skeleton model is at every moment and predict future poses by keeping track of the past positions of these joints.

II. Related Work

Applying *Human Action Recognition* (*HAR*) to mantain safety in the presence of moving robots is a popular approach, therefore, new ways to predict human intention are being constantly developed [2] [3] [4]. Previous work suggest two main ways of locating moving humans: vision-based methods [5] and sensor-based methods [6]. The main difference is the kind of equipment used to keep track of the human, in the first case human location is done using some kind of vision device such as a 3D-sensor while in the second case other methods are employed with devices such as accelerometers. However, as stated in [2] the sensor-based method is not practical in realistic environments because it requires the human to wear a special purpose suit. Nowadays safety access restriction mechanisms are the most extended solution for human-robot collaborative environments. Devices such as fences, single-plane laser scanning or proximity sensors are used to prevent the worker from entering the workspace during robot operation. However, as stated before, this reduces the cycle time of manufacturing processes whenever the robot needs to stop or reduce the working speed.

Previous work also found that tracking human's location in real time can guarantee the safety of a moving human applying two main methods: 1) proposed in [7] this approach defines a warning area around the complete body of the human and plans the trajectory of the robot so that it will never enter this warning area. 2) the alternative proposed in [8] is to plan the collision-free trajectory by early prediction of the location of the human. This work is built upon the latter approach contributing by applying early prediction to the location of specific joints of the human model instead of treating the human as a single object which enables a novel strategy to detect the initiation of the human action that will be presented later in this paper. Moreover, the architecture employed in this paper is designed to make it suitable to apply unsupervised

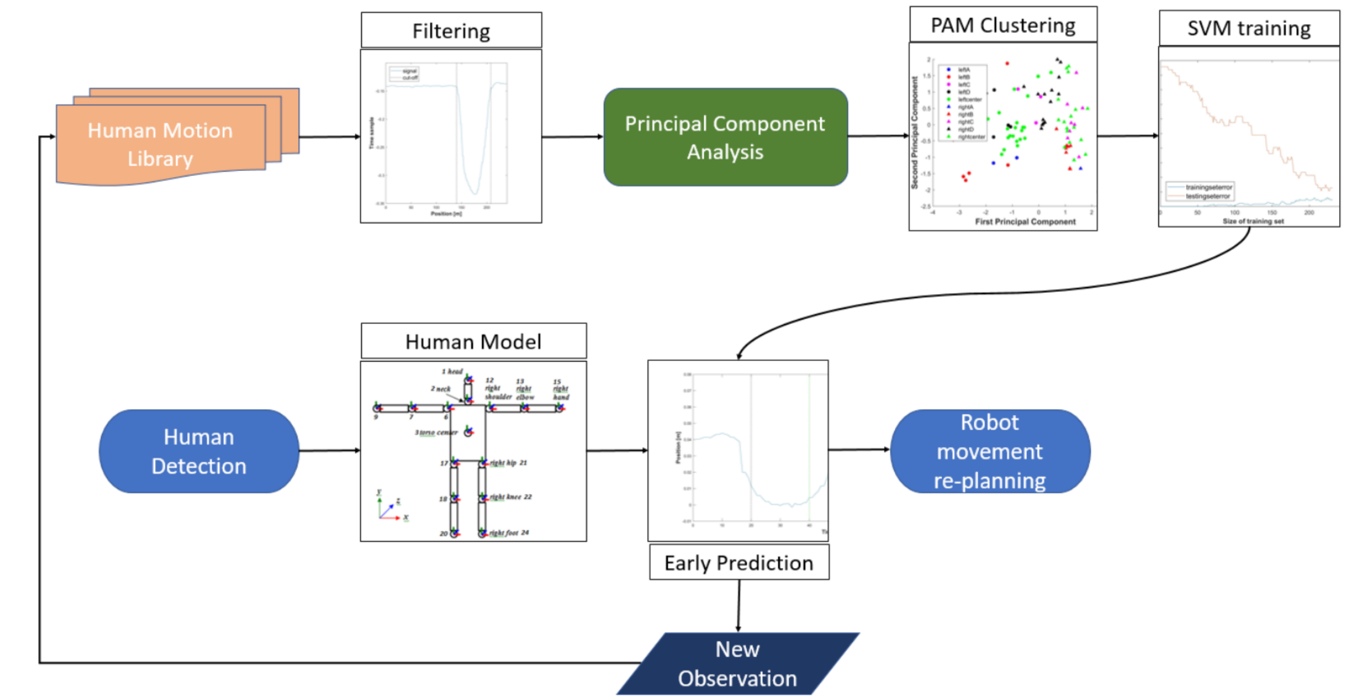

Fig. 2. Flow Diagram of the framework; the top branch represents the off-line phase with the data processing, clustering and classifier training; the bottom branch corresponds to the on-line phase were skeletal tracking and early prediction of human intention are performed.

learning techniques so that new classes can be added to the classifier without external intervention.

Regarding the early prediction, in [9], it is proposed to learn motion patterns using clustering and a *Hidden Markov Model* (*HMM*) while [10] followed a similar strategy but applying an *Input-Output HMM* (*IOHMM*). This approach is very interesting for early prediction of trajectories because it allows to train an individual *HMM* for each motion pattern and employ an *Expectation Maximization* (*EM*) algorithm to assign an observation to its corresponding pattern. In [9] they applied it successfully to improve the behaviour of mobile robots by predicting the motions of persons based on learned motion patterns. However it requires applying both *EM* and *HMM* to every single observation which can be more computationally expensive than other alternatives such as directly applying a classification algorithm to the data.

In [3], it is proposed to learn centers of clusters or trajectories using *PAM*, then they match to partial prototypes to form a hypothesis over future trajectories. The main goal in this paper was to calculate an edit distance which allowed to optimize the amount of produced prototypes, therefore improving the performance of the matching algorithm, which they achieved. The present work is using the same idea of clustering all of the observations into a limited amount of prototypical trajectories, however it uses *SVM* to classify new observations into the classes generated by *PAM* instead of trying to match prototypes. This approach should give the system more flexibility at classifying the observations.

In [4], it is argued that single-arm reaching motions for known tasks in collaborative settings are predictable. They use *Inverse Optimal Control* to learn a cost function that defines the movement of the human from the gathered trajectories. Afterwards, they predict motion for a given task by iteratively re-planning a trajectory for a human kinematic model using the STOMP algorithm with the learned cost function. This approach could be combined with an artificial potential field as suggested in [2] to allow the robot to avoid collision when the human follows a known sequence of tasks in normal operation while still safeguarding the human if it fails to follow the proper order. This approach is better suited to predict how a human will do a certain task in a collaborative environment than actually predicting which task is the human doing in real time. In the use case concerning this paper the idea of defining optimal trajectories for each action is used, however this optimal trajectories are inferred from the observations by using the prototypical trajectories for each class as optimal.

III. Method

This paper seeks a way for the robot to predict human intentions so as to better plan its own motion. By applying *PAM* to cluster the observations and *SVM* to classify new samples this work intends to enable the robot to improve predictions over time and be able to learn new movement patterns alone. The latter can be achieved by adding every new observation to the training dataset and re-running both *PAM* and *SVM* to retrain the model with the new samples.

Figure 2 shows the approach followed in this paper. The robot keeps a library with types of motions which it uses to train a classifier after pre-processing the raw data via filtering, *PCA* and clustering. Then, in real-time, it can use the centroid that represents the detected class as the most probable trajectory for the human to follow in the future. This enables the robot to move in a way less likely to interfere with the person. It also adds every new observation to the dataset so as to keep learning new examples.

Regarding the definition of the human actions, the workspace is considered 2-dimensional and a grid is defined over it so that reaching for a cell in the grid corresponds to a certain human action. This allows the classifier to be able to discriminate between human actions quite early and is easily extendible to form a 3-dimensional space in which a third coordinate will be accounted for. Each human action defined in this way is a separate class for the classifier. The main reason for defining the workspace as 2-dimensional in a first approach is because in this paper the main interest is to show that it is possible to train a classifier with this kind of set-up for which this definition of the classes is enough.

The approach to learning paths consists of using supervised learning (*SVM*) to classify normalized trajectories with similar characteristics represented as lists of timestamped coordinates. Normalization consists on taking a fixed amount of samples along the filtered trajectories so that each example has exactly the same amount of samples independently of its length or duration. A previous filtering and Principal Component Analysis (*PCA*) are also applied to the data to gather linearly uncorrelated samples from the collected coordinates and therefore perform dimensionality reduction. Further elaboration on these procedures can be found in Section *III.B*.

A. Data Acquisition

The experiment designed to gather training and test data consisted on a participant sitting in front of a table working on a sequence of individual tasks within the defined area. In order to execute the task, the participant must move one of the hands from the starting position to one of the various possible destinations and back, the exact sequence is randomly chosen and said out loudly for proper data labelling. The aim of the experiment is to simulate a pick and place task, for instance packing different chocolates into a sampler box.

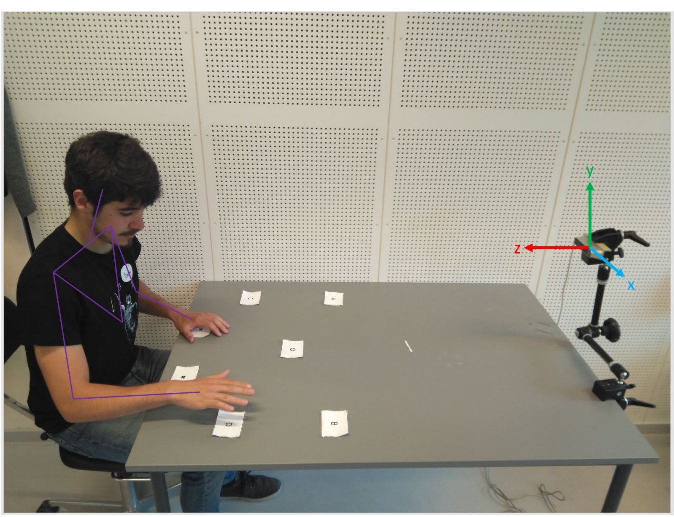

Fig. 3. Experimental set; in this Figure the 3-D sensor placed in front of the participant to avoid occlusions and the table with the grid defined to identify the different classes are visible.

Figure 3 depicts a participant taking part in the data acquisition. The positions on the table correspond to the 5 cells of the grid defined so that a total of 11 classes will be available for the classifier, one for each position and hand and one for the not moving case.

The main advantage of defining the classes like this is that it is easy to add new classes to test how well is the system able to learn patterns it has not seen before. Furthermore, the grid size can be reduced to achieve higher precision on the human action classification.

Recording method

In order to record the experiment, a Structure Depth Sensor is used. Using *OpenNI* tracker, a 15 joint model of the human body is obtained although only the upper body is accounted for. The camera is placed just in front of the participant to avoid any kind of occlusion during the experiment. It is fixed to the table so that the reference frame is always the same and the experiment is easily reproducible.

In Figure 3 the frame defined on the camera is visible, note how the y coordinate is of little help with the defined 2-D grid, however it could be used to extend the model so that a 3-D space could be used.

B. Data Processing

The raw data recorded during the experiments includes both the movement that is relevant for the *SVM* and a part were the operator is not moving. Consequently some method must be applied to perform this movement detection within the whole dataset.

In this paper data are filtered by thresholds around the median. Using the median as a reference is better than using the mean because it allows to detect the value of most of the data points corresponding to the part of the signal where there is no movement, while not being affected by the peaks corresponding to the actual movement.

Dimensionality Reduction

When handling as many variables as in this project, some kind of dimensionality reduction is advisable. All the trajectories are defined by 2700 features (9 joints x (3 coordinates per joint x (100 time samples)), however, many of these measure related properties and are therefore redundant. In this case *PCA* is applied. Let M be a subset of chosen eigenvectors that retain as much of the variance as required, λ be the corresponding eigenvalues and X_{mxn} be a matrix with m observations and n features. Then *PCA* allows to reduce dimensionality by:

$$M = X^T * M * (1 \oslash (\sqrt{m} \odot \lambda))^T \tag{1}$$

$$X = X * M \tag{2}$$

Note that in the equations (1) and (2), $\odot$ stands for element-wise product and $\oslash$ stands for element-wise division.

C. Classification

Use of a clustering algorithm allows the system to automatically add new observations to the dataset and therefore improve learning over time. The clustering algorithm is also used during the off-line training stage even if the labels of the data are already known. This leads to an increased error in the off-line phase but helps the clustering algorithm to properly cluster the new observed examples during the on-line stage.

In the off-line phase, *PAM* is used to generate sets of similar data points, afterwards, labels are assigned to these clusters by looking which class is most repeated within the cluster members. Then the dataset is divided into training, testing and cross-validation sets and the classifier is trained (Algorithm 1).

Algorithm 1 Classifier training procedure

1: **procedure** TRAINING(X_{init})
2: $X_{norm} \leftarrow$ NORMALIZE(X_{init})
3: $X_{pca} \leftarrow$ PCA(X_{norm}, n_{keep})
4: $Y_{pam} \leftarrow$ PAM($X_{pca}, n_{clusters}$)
5: $Y_{labelled} \leftarrow$ LABEL(Y_{pam})
6: $Model \leftarrow$ SVM($X_{pca}, Y_{labelled}$)
7: **return** $Model$
8: **end procedure**

The classifier must keep the parameters used to normalize and reduce the training set so as to allow the same procedure to be applied to new observations.

D. Prediction Method

The first step for the prediction strategy is movement detection, which consists on waiting until the camera provides a valid transform. Then data is acquired during a fraction of the time it usually takes to complete an action within the workspace and the classifier tries to detect which kind of human action is happening. If the observation is classified as *not moving*, the classifier discards that observation and tries again. Then, another set of data is recorded, it is appended to the previous one and a new prediction is attempted. In this way the algorithm is expected to keep improving the accuracy of the prediction as the participant gets closer to the destination. Once the time-out is reached the algorithm adds the recorded example to the dataset and starts over again. This procedure is depicted in Figure 4.

This sequential prediction strategy suggests that using several trained models may be advisable. Specifically a different model can be trained for every step in the recording sequence each of them trained with the same amount of time samples as there have been recorded at the corresponding point in the sequence.

In this work it has been found that both the presented strategy or an approach with a single classifier trained with all of the features at the same tame are able to perform predictions. This is due to the fact that currently each timestamped coordinate is considered as a single feature. Therefore an observation that has the first 10% of the features coincident with one of the classes will still be correctly identified by the classifier if the rest of the features are totally different from the rest of the available classes.

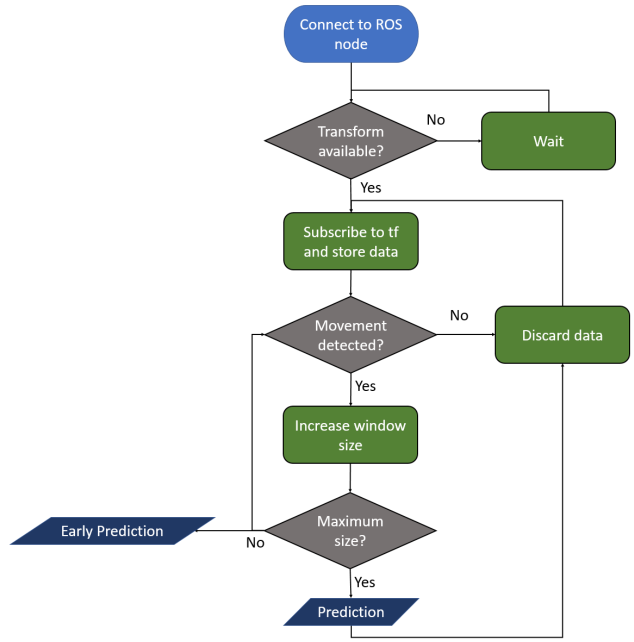

Fig. 4. Flow diagram of the prediction procedure; movement detection is triggered by any prediction different from the *Not moving* class; if movement is detected more samples are appended to the observation and prediction is attempted again.

IV. Results

A. Analysis of the Dataset

Figure 5 depicts the percentage of variability explained by the first 10 principal components, in total they explain up to 88% of the whole variance in the dataset. To avoid loosing more information than necessary, the dataset is truncated so that 99% of the total variability is retained for which 90 principal components must be used.

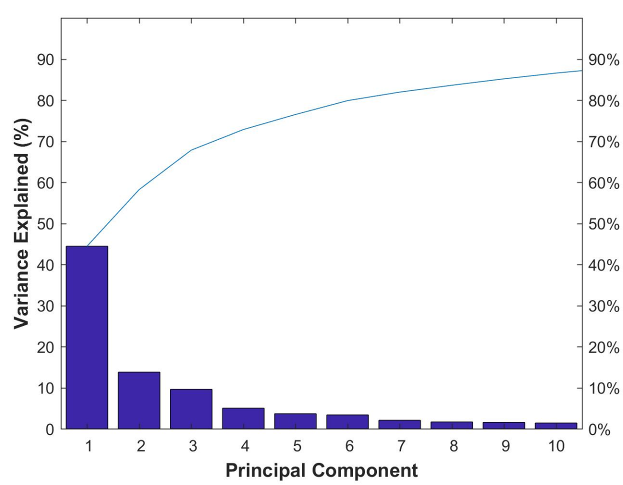

Fig. 5. Percentage of variability explained by the first 10 principal components; to avoid loosing to much information the first 90 principal components are chosen so that 99% of the variability is retained.

Afterwards a t-test can be applied to the reduced dataset in order to further limit the amount of features used by the classification algorithm which will reduce the execution time and allow a faster prediction. This test shows that just 30% of the features have p-values close to zero which means that less than 30 of the original 90 features have strong discrimination power. Using the 30 features left, Figure 6 uses *t-SNE* to reduce the dataset to just two dimensions and visualize it. It is noticeable that the data tend to group according to their classes which allows to expect a good performance of the classifier in this dataset.

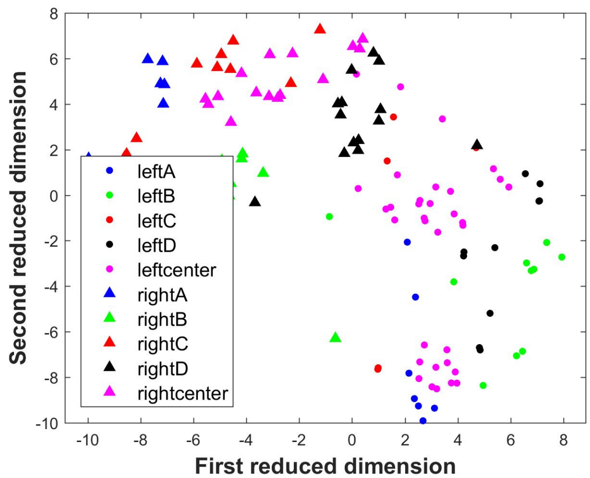

Fig. 6. Dataset visualized via *t-SNE*; the different classes are distinguishable, however some of them present more than one cluster (*lefcenter*), this does not pose a problem for the classifier because it works with more than the two dimensions used to depict the dataset; note the clear separation between left and right hand samples.

B. Model Evaluation

Training the algorithm with the resulting dataset shows a performance with accuracies around 80%, being the training set error about 10% and the testing set error around 15%. The relatively small difference between both errors suggests that the selection of features is working well on the dataset, however the relatively high bias implies that acquiring more training examples is likely to help improve the accuracy of the classifier.

The data acquisition in this paper is done tracking several human participants which means that the observations present differences within the same class which the algorithm needs to be able to neglect. This is the reason of the relatively high bias in the model.

C. Prediction

The dataset contains features that allow a proper classification of the movements. In order to show if early prediction is feasible, two strategies are compared. On one side the model is tested training it with with increasing windows from the starting part of the records, this will lead to several different models that the classifier will use sequentially as it keeps adding more time samples to the real-time observation. On the other side one single model is trained and it is used to classify these same windows.

Figure 7 represents how the accuracy of the algorithm varies depending on how much of the final trajectory is used for prediction for both strategies. Using several models shows better average performance but is strongly dependent on how well the beginning of the movement is detected.

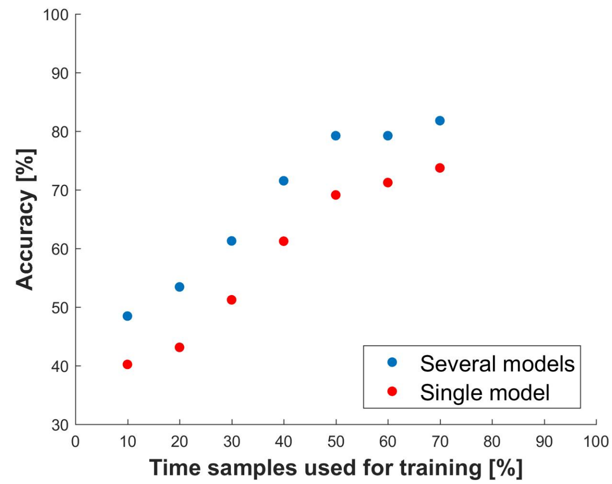

Fig. 7. Accuracy of the algorithm for early prediction; 40% of time samples corresponds to 1.2 seconds from a movement that takes a total of 3 seconds; over 70% time samples accuracies are not considered as it corresponds to the user going back to the starting position.

If the starting point of the new observation accurately coincides with the starting point of the ones in the training set, the strategy with several models performs better because it uses exactly the same amount of time samples as used during the training procedure. Using a single model, on the other hand, will stretch the reduced amount of time samples recorded through the whole duration of the observations in the training set so as to have the same amount of features which results in a higher overall error but lower dependence on how well the beginning of the human action is detected.

This paper found that the strategy followed to detect the beginning of the human action is accurate enough to allow a better performance of the several model strategy. With this strategy, an accuracy of around 70% can be reached with 40% of the total record being used for training and classifying. This accuracy can be improved by collecting more training examples.

the recording covers the full range from when the hands of the operator move from the reference position to when they go back to that position, this is intended to give the robot insight about the workspace occupancy during the whole movement. However for the prediction to be useful the robot needs to be able to know the intention of the operator before it gets to its final destination within the workspace.

In the current dataset it has been found that, on average, it takes the operator 3 seconds to get to the final destination and 1 to come back. Therefore the 40% time samples prediction point which was highlighted before means this system is able to predict the human intention within the first second of a movement that takes a total of 3 seconds.

V. Discussion

The strategy presented to detect when the operator initiates and concludes the movement was found to perform well with the considered set-up. Considering the use case, a sensor which detects an intrusion in the workspace can also be a good trigger for the prediction. Comparing the movement detection capability of the approach in this paper with a more traditional one based in a workspace occupancy sensor may be an interesting direction for future work. To this end it should be kept in mind that using the same strategy to detect beginning of operator motion both for the recording of the training dataset and the real-time observations is crucial to reduce unnecessary inaccuracies due to differences between the recording procedures.

Regarding the design of the experimental set. Recording the dataset so that the robot has information on the occupancy of the workspace also when the operator is going back to the starting position makes sense from the motion re-planing point of view. However, it does not help with the prediction. Retrieving separate datasets for prediction and motion re-planning can be a solution for this.

Another possible solution is to automatically detect when the operator reaches the final destination during the data processing stage, then the pre-processing algorithm should be able to cut the recorded observations so that only the first part is used for the prediction and the whole sample is used to the inference of the prototypical trajectory needed for the motion re-planning phase.

As for the amount of training examples acquired, even if more training examples will improve the performance of the classifier, they do so very slowly with every new observation that there is no point on collecting a bigger dataset within the scope of this project.

It is relevant to state that the strategy followed in this paper has been limited to predict a very reduced variety of movements within the workspace for simplicity. Nevertheless, it should be easy to define a grid in the workspace which allows to cover any possible movement within the area and therefore generalize this model. It will be interesting to see how well does this approach perform against the usage of regression for the same problem.

This work was mainly interested on analysing the feasibility of this strategy and these improvements are direction for future work.

VI. Conclusion

The use case of a pick and place task in a conveyor belt is a good candidate for learning operator intentions because of the easily monitorable workspace and the limited amount of possible tasks that can happen. Furthermore, this paper found that the approach of defining the classes for the classifier by a 2-D or 3-D grid fits very well in such a use case in which the human actions are very repetitive.

On another hand, the strategy followed in this paper to learn operator intentions has been shown to work well with the considered set-up. This strategy consisted on two clearly defined stages:

1) In the off-line stage a human motion library which defines as many off the possible human actions within the workspace as possible is gathered. Dimensionality reduction is applied to this human motion library to reduce the amount of features, which is crucial to allow a fast performance of the classifier and therefore make the possible predictions useful. Finally the resulting observations are clustered by an unsupervised algorithm and then a supervised classifier is trained.
2) In the on-line stage the pre-trained classifier uses the input data from a 3-D sensor monitoring the workspace to match the beginning of the newly observed human actions with the beginning of the ones in the training set and, therefore, perform a prediction.

The main goal in this paper was to show the capability of the strategies presented to perform human movement detection and early prediction of operator intentions. Defining a separate class which represents a standing human showed to be a good indicator to automatically detect human action initiation. On another hand, the results show that the prediction strategy provides accuracies comparable to those obtained using more complicated [2] [3] sets of features by just using the information about joint position during the whole trajectory and applying dimensionality reduction.

The usage given to the clustering algorithm also showed that it is possible to infer the different types of movements present in the dataset and, therefore, this strategy will allow the robot to automatically cluster new observations into the existing classes and keep improving its performance over time.

References

[1] F. Tobe "Why co-bots will be a huge innovation and growth driver for robotics industry" in *Automaton: IEEE Spectrum's award-winning robotics blog* consulted 05-19-2017.

[2] P. Zhang, P. Jin, G. Du, and X. Liu "Ensuring safety in human-robot coexisting environment based on two-level protection" in *Industrial Robot: An International Journal*, vol.43, lss 3, pp. 264-273, 2016.

[3] S. Xiao, Z. Wang and J. Folkesson "Unsupervised robot learning to predict person motion" in *IEEE International Conference on Robotics and Automation (ICRA)*, Seattle, Washington, 2015.

[4] J. Mainprice, R. Hayne, and D. Berenson "Predicting human reaching motion in collaborative tasks using inverse optimal control and iterative re-planning" in *IEEE International Conferece on Robotics and Automation (ICRA)*, Seattle, Washington, 2015.

[5] J. Krger, B. Nickolay, P. Heyer, and G. Seliger "Image based 3D surveillance for flexible man-robot cooperation" in *CIRP Annals - Manufacturing Technology*, vol. 54, no. 1, pp. 19-22, 2005.

[6] J. Corrales, F. Candelas, and F. Torres "Safe human-robot interaction based on dynamic sphere-swept line bounding volumes" in *Robotics and Computer-Integrated Manufacturing*, vol. 27, no. 1, pp. 177-185, 2011.

[7] J. Corrales, G. Garcia Gomez, F. Torres and V. Perdereau "Cooperative tasks between human and robots in industrial environments" in *International Journal of Advanced Robotic Systems*, vol. 9, no. 5, pp. 637-647, 2012.

[8] J. Mainprice and D. Berenson "Human-robot collaborative manipulation planning using early prediction of human-motion" in *The IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 209-306, Tokyo, Japan, 2013.

[9] M. Bennewitz, W. Burgard, G. Cielniak and S. Thrun "learning motion patterns of people for compliant robot motion" *The International Journal of Robotics Research*, vol. 24, no. 1, pp. 31-48, 2005.

[10] Z. Wang, R. Ambrus, P. Jensfelt, and J. Folkesson "Modeling motion patterns of dynamic obects by IOHMM" in *Intelligent Robots and Systems (IROS14)*, Chicago, Illinois, 2014.

[11] A. Mori, S. Uchida, R. Kurazume, R.-i. Taniguchi, T. Hasegawa and H. Sakoe "Early recognition and prediction of gestures" in *IPCR*2006.