Online Performance Prediction and Profiling of Human Activities by Observation

Emmanouil Hourdakis, Michail Maniadakis and Panos Trahanias

Abstract— **The capacity of a system to automatically analyze and predict the performance of a human in a particular task can provide important information in Human-Robot Interaction. Despite its usefulness, the above topic has received rather limited attention in the literature. In the current work, we introduce a method for performance prediction and profiling of human activities. Using little information about a task, our method is able to extract the characteristic motion patterns of an agent, analyze them and predict his/her performance in a given activity. We demonstrate the robustness of the method in several different activities, that involve both periodic and oscillatory primitive motions. In addition, we evaluate it thoroughly on data obtained from public datasets and discuss its usefulness for contemporary robotic applications.**

I. INTRODUCTION

Intelligent robots must be able to analyze human behavior, in order to handle demanding human-interaction scenarios [1]. However, this analysis is not always straightforward, since a person's behavior is highly dependent on context, environment and prior knowledge. For this reason, a considerable amount of research is devoted to issues that regard action interpretation [2], looking for methods to integate this information to models of human behavior. One important aspect that has not been investigated thoroughly in the relevant literature, is the ability to analyze and predict the performance of a human while executing a task. For robotic applications, this skill can prove invaluable, because it can furnish robots with the capacity to know when a human needs assistance, how to adapt their behaviors in order to be more productive or which human strategies are succesful so that they can be transferred to the robots.

Currently, performance analysis methods only exist for domains where the context is well defined and constrained. One typical example is sports [16][17][18], where recent research is developing new methods to automatically grade the performance of athletes. However, due to their task-dependent assumptions, these models rarely find use outside their application area. Aiming at more general performance evaluation approaches, it is important to develop methods that can quantify how well a human carries out a task without all the restricting assumptions imposed by a domain expert. This, however, presents an important challenge because physical and mental differences, prior experience, and different percepts result in across-subject and within-subject behavior variability. As a result, a standard template for behaviors can rarely be pre-defined, unless many assumptions are imposed.

In the current paper, we provide solutions to these problems, by introducing the Performance Prediction and Profiling Methodology (P^3M), to automatically *analyze* and *predict* the performance of an agent only by observation. P^3M requires little information about the task being analyzed. The behavioral analysis part of the methodology is handled automatically, by identifying and profiling the primitive motions that are characteristic of an agent's performance. As a result, models for new activities can be implemented easily. This makes the method readily applicable to a wide range of robotic application areas including:

- Domestic robots, allowing them to track the behavioral preferences and skills of humans over-time.
- Industrial setups, giving insights on how humans can improve their skills and abilities, and whether the use of robots could improve a certain workflow.
- Human-Robot Interaction scenarios, to facilitate productive cooperation with a human agent.

In the current paper we provide thorough examples on how to apply the methodology in two real life-activities, wiping a table, and tidying-up a table. We show how: (i) P^3M can confront scenarios, involving both discrete and oscillatory motions, (ii) components of the methodology can be readily transferred from one activity to another. In addition, we evaluate the ability of the methodology to predict the performance of a human, by applying it on a number of different activities found in public datasets. The main contributions of the current paper are:

- A unified method for performance analysis and prediction, that can be applied to both discrete and oscillatory motion patterns.
- The application of the method to several different activities, that involve everyday tasks, obtained from homemade and public datasets.

The rest of the paper is organized as follows. Section II reviews the relevant literature, while Section III offers a detailed presentation of P^3M. Section IV provides a detailed experimental evaluation of the proposed methodology and Section V concludes the paper and gives directions for future research.

II. LITERATURE REVIEW

Behavioral analysis is an important topic in human-robot interaction. One of its main challenges is to create simple

*This work has been partially supported by Greek National grand MIS 5030440 vipGPU and FETOPEN-4-2016-2017, Industrial Exploitation and Market Uptake of a Temporal Cognition Toolbox for Commercial Robots (ENTIMENT) .

The authors are with the Institute of Computer Science, Foundation for Research and Technology - Hellas (FORTH). ehourdak, mmaniada trahania@ics.forth.gr

and adaptive models, that can capture the complex nature of human behavior. Commonly, research is focused on probabilistic models, because they provide a framework for representing task uncertainty, dependencies, and temporal variation.

Related work on behavior modeling uses temporal models, such as Hidden Markov Models (HMMs) and Conditional Random Fields (CRFs), or causal models such as Bayesian belief networks, to create representative models of human behavior. HMMs can detect probabilistic regularities across sequences, and capture the structure of arbitrarily complex sequences given sufficient numbers of hidden states [3]. Due to the above, several works are employing them to create context-informed behavior models, including activity monitoring [4], anomaly detection [5] and multi-agent observation [6]. CRFs are discriminative latent variable models that have been shown to successfully learn the hidden structure of a problem domain [8]. They define a conditional probability distribution over label sequences, thus lifting the independence assumptions required by the HMMs [9]. Both models can be discussed as special cases of dynamic Bayesian networks, i.e. probabilistic networks that represent a sequence of temporal variables. The Bayesian formulation provides a more general framework for behavior monitoring [20], because it can be used as a generic model for handling both uncertainty and incomplete data.

Supervised activity models, such as the ones above, are based on the assumption that there exist well-defined and known a priori behavior classes. However, this is not always the case in real-life interaction scenarios. For this reason, a number of techniques have been proposed for unsupervised learning of behavior, including [7][8]. The benefit of these methods is that they don't assume any structure on the data. In [10] the authors use latent topic models, such as the Probabilistic Latent Semantic Analysis model to derive action categories. Jurek at al [11] uses a similar approach where behaviors are profiled using unsupervised methods and later rectified through data mining techniques.

All the models discussed focus on profiling behaviors for the sake of recognition and classification tasks. Currently, there is very little work to investigate the effect that these behaviors have on the activity itself. This type of performance profiling can only be found in sports, where the activity and performance measures are well defined. For example in [16] the authors use a visual feed to score the events of Olympic sports using spatiotemporal features learned from a 3D convolutional neural network. Similarly, in [17], they judge scores to rhythmic gymnastic movements by transforming movements into specialized spatiotemporal image templates. In [18] they attempt to generalize the problem domain by proposing the Segment-based P3D-fused network, that learns action quality scores from previous datasets. Finally, in [19] the authors suggest a scoring system to evaluate performance based on comparisons of different measures.

The aformentiond methods are task specific, and cannot be applied outside their problem domain. To employ behavior profiling in Human-Robot interaction, it is important to have methods that can perform on a broad spectrum of tasks. In the current paper, we address this need by introducing a methodology that can derive performance profiles, assuming only a univariate measure for the task progress. P^3M automatically finds the behaviors that are characteristic of a person's performance, while it automatically assigns perfomance indicators for each primitive internally. In addition, its probabilistic formulation allows predicting the performance and progress of a human with low computational cost and robustly. As a result, the method is readily applicable to a wide range of robotic scenarios.

III. Methodology Formulation

In the current section, we outline the mathematical formulation of P^3M, and also study its application in activities that involve both oscillatory and discrete actions. There are three components in the methodology:

- *Activity Observation* used to extract the raw motion and task progress information, by observation.
- *Profiler* used to create behavioral profiles.
- *Performance Prediction*, that employs the output of the previous two components and derives predictions for the user performance and task progress.

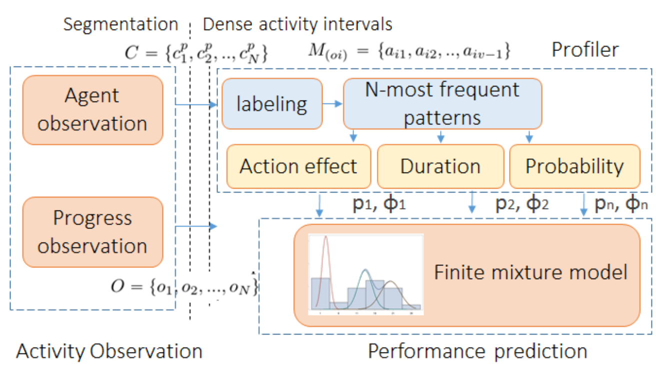

Fig. 1: P^3M block diagram highlighting the three components of the methodology, and the information they manage.

Figure 1 provides a block diagram of P^3M highlighting its three main components and inter-connections across them. In the rest of this section we detail the formulation of the above three components, using the task of tidying a table as a running example. In this task, a human operator tidies up a table by moving objects that are scattered all over the table surface to their predefined store locations, on the side of the table. The whole task is being observed by a camera overlooking the scene.

A. Activity Observation

An activity is analyzed using two streams of information: *Agent* and *Progress* observation.

Agent Observation: The first is responsible for observing the agent, while performing the activity, and extracting raw motion information. In the current formulation, we use mean shift tracking [13], to track the end-effector of the human performing the activity. The tracking data are stored

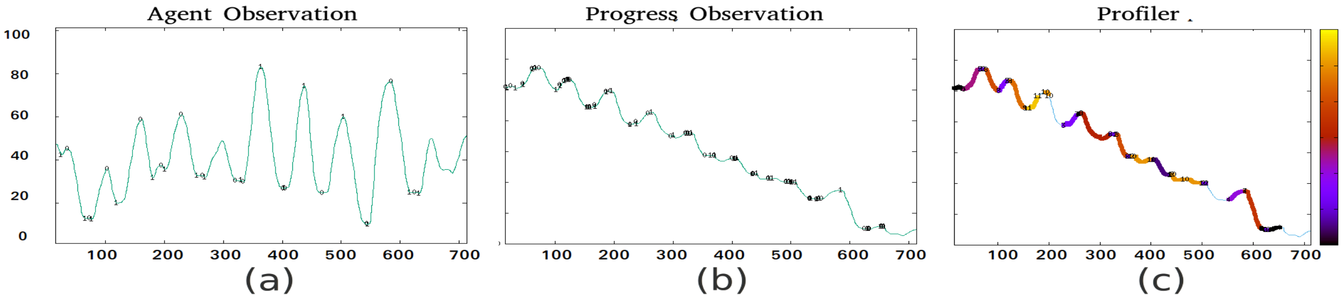

Fig. 2: Primitive extraction for the tidy the table task. (a) Primitives extracted by the *Agent Observation* component; (b) Segments extracted by the *Progress Observation* component; (c) Primitives labeled by the *Profiler*. The x-axis indicates time-steps of the experiment. The y-axis indicates the 3D distance from the table's bottom-right corner for (a), and the normalized task progress for (b) and (c). The colored segments in (c) indicate the primitives labeled by the Profiler.

as $C = \{c_1^p, c_2^p, .., c_N^p\}$, a set of p dimensional vectors corresponding to the state variables that characterize the observed activity. For the tasks we discuss, C includes the 3D coordinates x, y, z of the hand position at time-step i, i.e. $c_i^3 = \{x_i, y_i, z_i\}$.

Progress Observation: This is the only module that must be set for implementing a new activity. It essentially requires the definition of a univariate measure for the task progress, in the form of $O : \mathbf{t} \mapsto \mathbb{R}$, with $O = \{o_1, o_2, ..., o_N\}$ a univariate time-series, uniformly sampled at regular intervals. In the results section we describe potential implementations of this component for various different tasks.

Segmentation: The last module of the activity observation component segments the two observation signals densely, in order to obtain a rich set of activity intervals. It smooths the signal's first derivative and locates the corresponding zero-crossings. To grade each local extrema it employs the prominence algorithm [14], which calculates a significance value for each. We keep the n largest peaks whose significance value exceeds a certain threshold. Figures 2(a) and 2(b) show the agent and progress observation signals, for the tidy the table task, segmented using the above process.

The segmentation process yields a set of n non-regular intervals $I_c = \{\{t_{c_{s1}}, t_{c_{e1}}\}, \{t_{c_{s2}}, t_{c_{e2}}\}, .., \{t_{c_{sn}}, t_{c_{en}}\}\}$, where all values in I_c correspond to the start t_{c_s} and end times t_{c_e} that the segmented motions are observed in C. Similarly, an $I_o = \{\{t_{o_{s1}}, t_{o_{e1}}\}, \{t_{o_{s2}}, t_{o_{e2}}\}, .., \{t_{o_{sn}}, t_{o_{en}}\}\}$ point set is obtained for the O progress observation. The two sets are merged into one dense new set I_u, that contains all time segments in the activity where a change in activity progress or human motion is detected.

B. *Profiler*

The *Profiler* is responsible for processing the motions in the dense segments I_u, in order to identify the n-most frequent behavior patterns of the agent. This occurs in two steps: (i) labeling the intervals in the I_u set, (ii) processing the labeled sequence, with a pattern mining algorithm, in order to find the most frequent patterns.

1) Labeling: To obtain the behavior labels, the profiler assigns a vector $M_{(oi)} = \{a_{i1}, a_{i2}, .., a_{iv-1}\}$ which contains all motion related information. The n behaviors in I_u are classified into K clusters, by maximizing the intra-class variation in the $M_{(oi)}$ feature vector:

$$W(C) = arg\max\Big(\sum_{i=1}^{K}\sum_{j=1}^{n}\|\mathbf{M_{(i)}} - \mathbf{M_{(j)}}\|\Big) \qquad (1)$$

where $W(C)$ is the intra-class variation. K is set by the system operator at a low value for rather simple tasks, and at higher values for more complex tasks. Higher K values will result in more behaviors being profiled and additional computational resources. From eq. 1 we obtain the K different cluster centers $\{M_{(1)}, M_{(2)}, .., M_{(K)}\}$ that are used as labels for the different primitive actions appearing in the activity. In the current implementation, eq. 1 is solved using k-means, but any other clustering method would also suffice.

After labeling the set I_u, we obtain a sequence of labels that correspond to the behavioral patterns that were observed in each segment. The profiler processes these labels with prefix span [15], a sequential pattern mining algorithm, in order to look for the n-th most frequent sequences. These are then assigned new labels, as frequently observed behaviors, and are used to create the profile of the human. The following figure illustrates this process (Fig. 3).

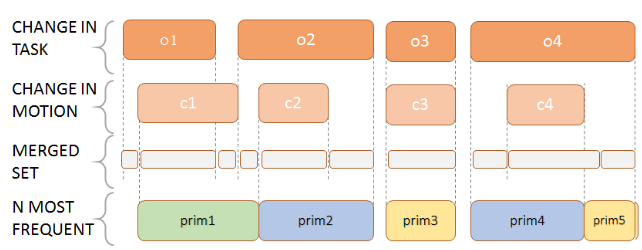

Fig. 3: Merging and synthesizing the segmentation intervals by the profiler. C indicates the intervals obtained by the *Agent Observation* component. O indicates the intervals obtained by the *Progress Observation* module. The merged intervals are marked with grey, while the re-synthesized intervals of the n-most frequent patterns are shown in last row.

This process effectively allows us to profile both discrete and oscillatory behaviors. In both cases it will merge motions in order to find the most frequent patterns of the agent. This is evident from Fig. 2(c), which shows the motions selected for an activity that involves both discrerte and oscillatory motions.

The next step of the profiler is to assign performance indicators for the primitives identified. Three quantities are calculated for each primitive: *(1)* the effect that it has on the task progress, *(2)*, its frequency of occurrence within C, and (3), its duration. The latter is trivially obtained by subtracting the start time from the end time of each interval in the I_c and I_o sets. Below we describe how the former two are derived.

Effect of each primitive: To describe how each primitive affects the activity progress, the profiler assigns a function $f_{M_{(i)}}(t)$ to each $M_{(i)}$, for which it holds:

$$O(t_{pe}) = O(t_{ps}) + \int_{t_{ps}}^{t_{pe}} f_{M_{(i)}}(t)dt, \quad \forall\{t_{ps}, t_{pe}\}\epsilon I^{M_{(i)}} \tag{2}$$

i.e. the change in task progress during the interval $\{t_{ps}, t_{pe}\}$, where primitive $M_{(i)}$ is observed, can be determined by the integral of the function $f_{M_{(i)}}$ for that time period. P^3M approximates the function $f_{M_{(i)}}$ automatically using the output of the function $O(t)$, provided by each task implementation.

Frequency of occurrence: To calculate the probability pr_i, the profiler uses Kernel Density Estimation (KDE). Given a sample set of n values from an identically distributed variable l, the Density estimator p around a point l_0 is:

$$p(l_0) = \frac{1}{n}\sum_n G_h\Big(\frac{l-l_0}{dl}\Big) = \frac{1}{nh}\sum_n G\Big(\frac{l-l_0}{hdl}\Big) \tag{3}$$

where $G(z,\sigma) = \frac{1}{\sqrt{2\pi}\sigma}e^{-\frac{z^2}{2\sigma^2}}$ is the Gaussian kernel, and h is the Kernel bandwidth, which acts as a smoothing parameter. For the task examined in this paper, KDE suffices for representing task complexity. In cases where one would like to analyze a task with structural hierarchy, this step could be replaced with temporal probabilistic approaches such as HMM.

C. Performance prediction

To predict the performance indicators, the method uses a finite mixture approach to approximate a function $f : \mathbf{C}, \mathbf{t} \mapsto \mathbb{R}$, that describes how each behavior affects the progress of the activity:

$$O(t) = f(C,t) = \sum_{i=1}^{\xi} p_i\phi_i(C,t) \tag{4}$$

where p_i are the mixing components, that satisfy $p_i \geqslant 0 \forall i\epsilon\xi$ and $\sum_{i=1}^{\xi} p_i = 1$, ϕ_i are the local basis functions as described below. To obtain ϕ_i we use the new set of intervals described in section *D*, through which we obtain the $\{M_{(1)}, M_{(2)}, .., M_{(K)}\} \rightarrow \{f_{M_{(1)}}, f_{M_{(2)}}, .., f_{M_{(K)}}\}$, that describe how each primitive changes the task progress. Using equations 3, 4 one can derive useful information about the activity: Given the probability $p(i)\forall i\epsilon k$ for all primitives one can estimate, using eq. 4, how the task progress will change from t to t_k:

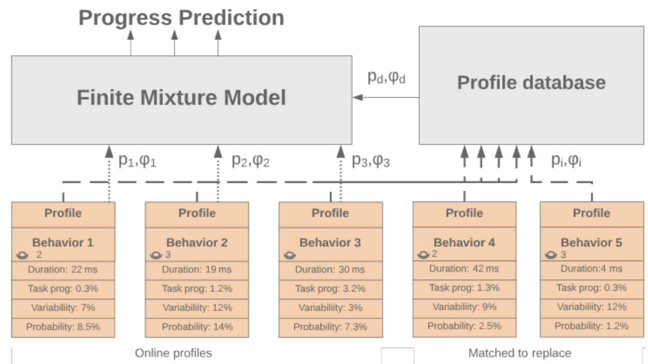

Fig. 4: Predicting the performance using the profiler.

$$O(t+t_k) = O(t) + \sum_{i=1}^{k}\left(p(i) * \int_{t}^{t+d_i} f_{M_{(i)}}(t)dt\right) \tag{5}$$

Eq. 5 provides an estimate of the activity progress forward in time, using the $f_{M_{(i)}}$ as basis functions. t_k is the expected duration for all primitives, i.e. the duration of a primitive, scaled by its probability, and is calculated as $t_k = \sum_{i=1}^{k}(p(i) * d_i)$, where d_i is the average duration for a certain primitive, and $p(i)$ is its probability.

Finally, the method can also take advantage of stored user profiles, in cases where this information is available. When a profile contains primitives that are not frequent, P^3M can search stored profiles and replace them with behaviors from a similar profile for the same task. To find the most similar profile, P^3M looks at the intra-class distance $\|\mathbf{M_{(i)}} - \mathbf{M_{(j)}}\|$ for each stored primitive. The most frequent behaviors from the matched profile are used to replace the rarely observed primitives in the Finite Mixture Model. This is illustrated in Fig. 4, where two of the least observed primitives are replaced with ones from a profile database.

IV. RESULTS

In the current section, we present evaluation results of P^3M, when applied to a number of tasks obtained from homemade and public datasets. The results fall into two categories. The extraction of performance indicators for each activity and the predictions regarding this performance.

A. Application on two household activities

1) Wipe the table: This task refers to wiping the surface of a table using a sponge. The *Activity Observation* tracks the percent of the surface of the table that has been wiped. To extract the surface of the table, we employ a model-based method, which detects the table surface and its boundaries. Then, the activity progress is determined by tracking the position of the sponge used in the task, and estimating the percent of the table that has been wiped, e.g. $o_N = 30$ means that the table has been wiped by 30% at time t_{o_i}.

2) Tidy the table: This task refers to tidying a table, i.e. storing the objects clutered on a table to their designated locations. The progress is determined by the sum of the distances of all objects to their store location. To determine it, we use mean-shift tracking to track the position of all objects and measure their relevant Euclidean distances.

We evaluated P^3M by using it to profile the performance of 10 different people over 100 trials. First we assesed whether it can predict the progress of the task, in different time-steps of the experiment. Table I summarizes the prediction accuracy for 100 trials of the two tasks, performed by 10 users.

TABLE I: Evaluation of P^3M on two activities: (i) Wipe the table and (ii) Tidy the table. P^3M was used to predict the task progress at different time-steps (300, 500 and 800 time-steps).

Task	**P^3M Progress**	**Actual Progress**	**Error**
Wipe the table			
at time-step 300	54 %	43%	9%
at time-step 500	65%	58%	7%
at time-step 800	96%	100%	4%
Tidy the table			
at time-step 300	45%	52%	9%
at time-step 500	56%	72%	8%
at time-step 800	84%	89%	5%

As Table I demonstrates, the methodology can predict the progress that a human will have on a task, with high accuracy. Results reported for different time-steps of the experiment have errors less than 10%. Predictions are more accurate after observing the activity for a while, (e.g. at time-step 800), because sufficient information has been gathered by the method.

For the first task, the method identified 12 most frequent primitives used by the agent, and reported their corresponding performance metrics, as shown in Fig. 5.

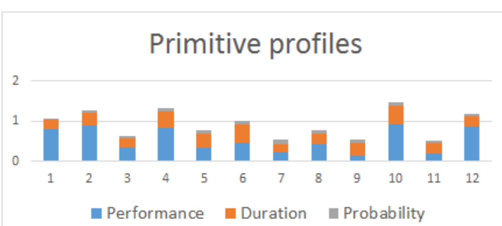

Fig. 5: Performance information collected for the 12 most frequent primitives, identified for the wipe the table experiment.

For each of the observed patterns the method was able to estimate its probability, duration and performance. Based on this information P^3M was also able to estimate the overall efficiency of a human in the task. For each primitive, this measure is calculated as the fraction of the performance divided by duration and scaled by its probability. The overall task efficiency is normalized in the [100-0] range across all primitives (Fig. 6).

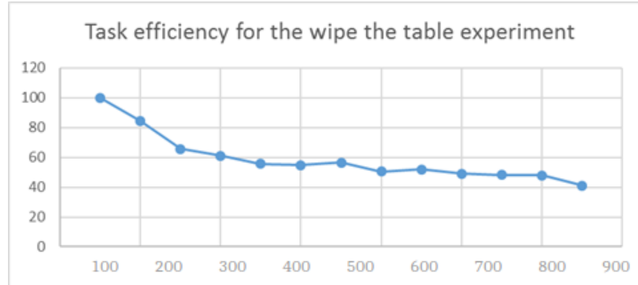

Fig. 6: Efficiency for the wipe the table experiment, normalized in the [100-0] range, recorded at 13 different time instants of the activity. The x-axis indicates the time-steps of the experiment.

As can be seen from the figure, the task efficiency for the wipe the table experiment, starts with a high value and later reduces over time. Lower efficiency values indicate that primitives have reduced performance and larger durations. This is expected for the wipe the table task because at later stages of the experiment, large parts of the surface have already been wiped. When the sponge runs over those regions, the activity progresses slower.

P^3M was also able to provide predictions on the user's future performance, using the equations described in Section III-C. Results are shown in Fig. 7, where it is evident that P^3M can predict the progress of the task with high accuracy. In the initial steps, where only a few data have

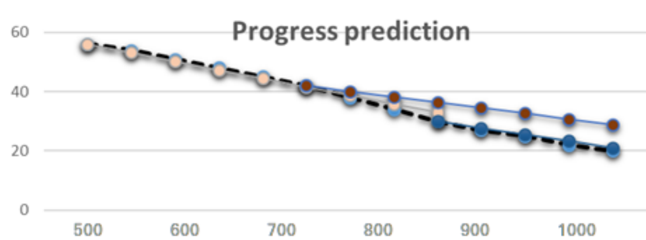

Fig. 7: Predictions for the future task progress in three time-steps during the wipe the table experiment: At time-step 500 (pink circles), at time-step 725 (red circles) and at time-step 870 (blue circles). The actual task progress is shown with a dashed ciel line. The x-axis indicates the experiment time-step, while the y-axis the % of the task progress.

been obtained, the quality of these predictions was rather low. However, as more primitives were observed and analyzed, the methodology was able to predict more accurately the task progress.

For the second task, tidy the table, 10 most frequent primitives where observed, as shown in Fig. 8. These primitives were used to predict the task progress, based on the human's performance, as shown in Fig. 9. As the figure shows, P^3M's prediction is accurate despite the fact that a large fraction of the primitives observed had low repeatability. Using this information P^3M can track the performance of a human over multiple trials (Fig. 10), providing a useful insight on how one's skills evolve over time.

TABLE II: Evaluation of the methodology on activities found in public datasets.

Task	P^3M **Progress**	**Actual Progress**	**Error**	**Progress model**
Empty dishwasher (P19-emptydishwasher-ch0) [22]	63%	72%	9%	# of utencils removed
Clean the floor (P23-sweep-ch0) [22]	45%	53%	8%	% of the floor that has been wiped
Setup the table (P16-settable-ch0) [22]	87%	96%	9%	# of utencils placed on the table
Cut three ingredients (s10-d02-cam-002) [21]	56%	63%	9%	cm of each ingredient cut
Garniture pizza (s15-d07-cam-002) [21]	82%	89%	7%	# of tomatoe slices placed
Chope vegetables (cut_cucumber_core) [23]	53%	42%	9%	cm of vegetables choped

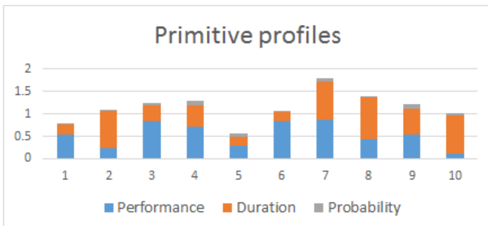

Fig. 8: Performance information collected for the 10 most frequent primitives, identified for the tidy the table experiment.

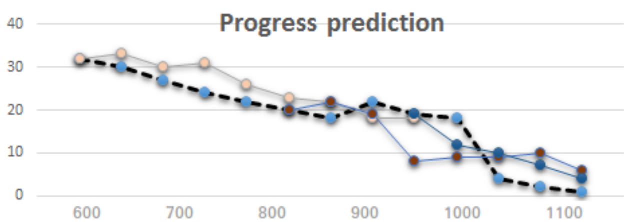

Fig. 9: Predictions for the future task progress in three time-steps during the tidy the table experiment: At time-step 600 (pink circles), at time-step 820 (red circles) and at time-step 950 (blue circles). The actual task progress is shown with a dashed ciel line. The x-axis indicates the experiment time-step, while the y-axis the % of the task progress.

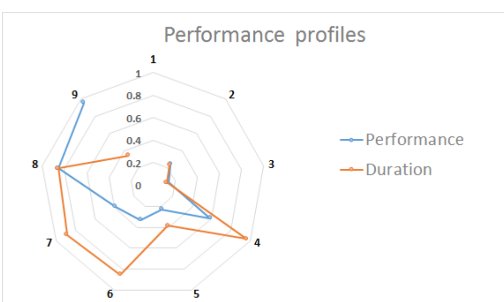

Fig. 10: Performance information for 9 different trials of a user, showing how his/her average duration (orange line) and performance (blue line) evolve over consecutive trials.

B. Evaluation on public datasets

We further evaluate P^3M using six different activities, obtained from the following public datasets: (i) MPII cooking dataset [21], (ii) The KIT Robo-Kitchen Activity Data Set [22], and the (iii) 50 salads dataset [23]. The videos selected focus on household activities, and cover a broad spectrum of tasks, ranging from floor cleaning to vegetable cutting (Fig. 11). For each task, we implemented a progress model, which is summarized in Table II, right most column.

Fig. 11: Six Daily activities, found in public datasets, used to evaluate the methodology. From left, top: (i) chopping vegetables, (ii) cleaning the floor, (iii) setting up a table, (iv) cutting three ingredients, (v) emptying a dishwasher, (vi) putting garniture on a pizza.

As Table II indicates, the methodology can predict the progress of a human on all the selected tasks with high accuracy. Performance indicators could be extracted relatively easy for each task, using a simple definition for the progress observation model of each activity.

V. CONCLUSIONS

In the current work, we address an important capacity of rorbotic systems, by introducing a performance prediction and profiling methodology for everyday Human-Robot interaction scenarios. P^3M is fast, and can be easily implemented for a number of different activities. Extensive experimentation in real scenarios shows that P^3M can extract accurate measures of the user's performance.

Various robotic disciplines can be benefited from the proposed methodology, including service and industrial robots. For example, P^3M could inform a robot when a human needs assistance, or whether a robot can be useful in a certain scenario. In more complex activities P^3M could even be used by a robot to provide specialized assistance, by helping a human only where he lacks the skills to perform an activity. In the future, we plan to extend the methodology with methods that automate the task of defining the progress of the activity. To this end, we will investigate how Deep Learning methods can be used to automatically extract task representations.

REFERENCES

[1] Leite, Iolanda, Carlos Martinho, and Ana Paiva. "Social robots for long-term interaction: a survey." International Journal of Social Robotics 5.2 (2013): 291-308.

[2] Aggarwal, Jake K., and Lu Xia. "Human activity recognition from 3d data: A review." Pattern Recognition Letters 48 (2014): 70-80.

[3] Chung, Pau-Choo, and Chin-De Liu. "A daily behavior enabled hidden Markov model for human behavior understanding." Pattern Recognition 41.5 (2008): 1572-1580.

[4] Debes, Christian, et al. "Monitoring activities of daily living in smart homes: Understanding human behavior." IEEE Signal Processing Magazine 33.2 (2016): 81-94.

[5] Ishii, Haruka, et al. "Method of behavior modeling for detection of anomaly behavior using hidden markov model." 2018 International Conference on Electronics, Information, and Communication (ICEIC). IEEE, 2018.

[6] X.H. Liu, C.S. Chua, Multi-agent activity recognition using observation decomposed hidden Markov models, Image and Visual Computing 24 (2) (2006) 166175

[7] Yu, Tsz-Ho, and Yiu Sang Moon. "Unsupervised Abnormal Behavior Detection for Real-Time Surveillance Using Observed History." MVA 2.8 (2009): 76-80.

[8] H. Zhong, J. Shi, and M. Visontai, Detecting Unusual Activity in Video, Proc. IEEE Conf. Computer Vision and Pattern Recognition, pp. 819-826, 2004.

[9] O. Boiman and M. Irani, Detecting Irregularities in Images and in Video,Proc. 10th IEEE Intl Conf. Computer Vision, pp. 462-469, 2005

[10] Niebles, Juan Carlos, Hongcheng Wang, and Li Fei-Fei. "Unsupervised learning of human action categories using spatial-temporal words." International journal of computer vision 79.3 (2008): 299-318.

[11] Jurek, A., Nugent, C., Bi, Y., and Wu, S. (2014). Clustering-Based Ensemble Learning for Activity Recognition in Smart Homes. Special Issue of MDPI Sensors Journal for UCAmI IWAAL, 14:1228512304.

[12] Moutacalli, Mohamed Tarik, Abdenour Bouzouane, and Bruno Bouchard. "The behavioral profiling based on times series forecasting for smart homes assistance." Journal of Ambient Intelligence and Humanized Computing 6.5 (2015): 647-659.

[13] Henriques, Caseiro, Martins, and Batista] Joo F Henriques, Rui Caseiro, Pedro Martins, and Jorge Batista. Exploiting the circulant structure of tracking-by-detection with kernels. In European conference on computer vision, pages 702715. Springer, 2012.

[14] G Palshikar et al. Simple algorithms for peak detection in time-series. In Proc. 1st Int. Conf. Adv. Data Analysis,

[15] Pei, Jian, et al. "Prefixspan: Mining sequential patterns efficiently by prefix-projected pattern growth." icccn. IEEE, 2001.

[16] Diaz-Pereira, M. P., Gomez-Conde, I., Escalona, M., Olivieri, D. N. (2014). Automatic recognition and scoring of olympic rhythmic gymnastic movements. Human movement science, 34, 63-80.

[17] Khong, Serena WJ, and Pui W. Kong. "A Simple and Objective Method for Analyzing a Gymnastics Skill." European Journal of Physical Education and Sport 2 (2016): 46-57.

[18] Xiang, X., Tian, Y., Reiter, A., Hager, G. D., Tran, T. D. (2018, October). S3d: Stacking segmental p3d for action quality assessment. In 2018 25th IEEE International Conference on Image Processing (ICIP) (pp. 928-932). IEEE.

[19] Doughty, Hazel, Dima Damen, and Walterio Mayol-Cuevas. "Who's Better? Who's Best? Pairwise Deep Ranking for Skill Determination." IEEE Conf. on Computer Vision and Pattern Recognition. 2018.

[20] Xiao, Q., and Song, R. (2018). Action recognition based on hierarchical dynamic Bayesian network. Multimedia Tools and Applications, 77(6), 6955-6968.

[21] Rohrbach, Marcus, et al. "A database for fine grained activity detection of cooking activities." IEEE Conf. on Comp. Vision and Pattern Recognition, 2012.

[22] Rybok, Lukas, et al. "The kit robo-kitchen data set for the evaluation of view-based activity recognition systems." 2011 11th IEEE-RAS International Conference on Humanoid Robots. IEEE, 2011.

[23] Stein, Sebastian, and Stephen J. McKenna. "Combining embedded accelerometers with computer vision for recognizing food preparation activities." ACM international joint conference on Pervasive and ubiquitous computing, 2013.