2020 IEEE International Conference on Robotics and Automation (ICRA)
31 May - 31 August, 2020. Paris, France

PARC: A Plan and Activity Recognition Component for Assistive Robots

Jean Massardi[1], Mathieu Gravel[1] and Éric Beaudry[1]

***Abstract*— Mobile robot assistants have many applications, such as helping people in their daily living activities. These robots have to detect and recognize the actions and goals of the humans they are assisting. While there are several widespread plan and activity recognition solutions for controlled environments with many built-in sensors, like smart-homes, there is a lack of such systems for mobile robots operating in open settings, such as an apartment. We propose a module for the recognition of activities and goals for daily living by mobile robots, in real time and for complex activities. Our approach recognizes human-object interaction using an RGB-D camera to infer low-level actions which are sent to a goal recognition algorithm. Results show that our approach is both in real time and requires little computational resources, which facilitates its deployment on a mobile and low-cost robotics platform.**

***Index Terms*— Activity recognition, Plan recognition, Computer vision, Robotic assistant, Activities For Daily Living, Object affordance, Particle filter**

I. INTRODUCTION

Robot assistants bring interesting perspectives to assisting people in their Activities for Daily Living (ADL), especially people with cognitive or physical disabilities. Pearl [1] and Care-o-bot [2] are examples of assistant robots that show that such systems could improve the quality of life at home for seniors. One of the advantages of robot assistants over other technologies, like smart-home environment, is that they can work with on-board sensors in open spaces and not only in controlled and monitored environments. This means that robot assistant tend to be less intrusive than installing many built-in sensors everywhere and can have a lower cost.

To perform assistive tasks, the robot has to recognize what a person is currently doing, what they will do next and what their intents are. This problem corresponds to plan and activity recognition problems[3]. Predictions with increased details give better information to the robot, which tends to improve its ability to perform assistive tasks.

At present, robot assistants do not usually perform plan recognition but focus on activity recognition [4]. For a robot assistant to perform plan recognition, the plan recognition approach has to be: (1) expressive enough to represent complex plans; (2) quick enough to be used in real-time constraints; and (3) able to handle noisy observations. Until recent work, some plan recognition approaches could handle these problems separately [5] [6] [7] [8], but not all of them at the same time. Work from Massardi et al. [9] proposes an approach to plan recognition that can handle all these constraints, making it a good candidate for integrating plan recognition into a robot assistant.

This article presents PARC, a Plan and Activity Recognition Component for assistive robots. PARC can handle noisy observations and has light requirements for computational resources, making it particularly well suited for robotics assistant platforms. To do so, PARC represents low-level actions as interactions between a person and objects, recognizes these actions using an activity recognition layer that relies on object recognition, and recognizes the goal and future probable actions using a plan recognition layer. All of these are done online while using limited sensors and resources. This is a boon for the robotic assistants that must offer a quick reaction time to offer appropriate aid and/or feedback to the person.

This paper is organized as follows. Section II provides a background of activity and plan recognition systems. Section III describes the PARC framework. Section IV shows the implementation of the system on a prototype and the results based on the recognition rate of the user activity over our own plan corpus. A conclusion follows.

II. RELATED WORK

This section focuses on two major concepts related to PARC: (1) current approaches to activity recognition for robotic companions tasked to air during ADL; and (2) studies in plan recognition.

A. Human activity recognition

At its base, human activity recognition is the detection and recognition of sets of actions that are attributed to a specific activity. To do so, many existing approaches extract descriptors from the environment along with human pose estimations to represent actions[10].

One popular method for activity recognition is using sensor-driven recognition to model specific activities[11]. Linner [12] got a 93.3% recognition rate for ADL recognition using Radio Frequency IDentification (RFID) tag triangulation. While these methods show high recognition accuracy rates, the use of additional sensors for every possible human-object interaction limits the adaptability of the approaches for new activities that would need new sensors.

The other main type of activity recognition is vision driven. Instead of using multiple sensor data, the agent uses visual cues as input, such as joint and pose estimations on learned deep network models to recognize activities [13] [14]

*This work was supported by AGE-WELL and NSERC.

[1]Authors are with the Department of Computer Science of the Université du Québec à Montréal, Canada. gravel.mathieu.3@courrier.uqam.ca, massardi.jean@courrier.uqam.ca, beaudry.eric@uqam.ca

[15]. Xia et al. [15] achieved a 82% recognition rate for activity recognition using a Hidden Markov Model (HMM) based on reduced shape and motion features extracted from the user pose estimation. These methods rely on pre-learned activities models, which need a robust training set to take into account the various ways and possible set of actions different users can take to execute their ADL.

Recent works in the domain propose the usage of affordances to model ADL activities as human-object interactions[16]. The concept of affordance is a constant field of study in the robotics and computer vision scientific community. Montesano et al.[17] describe it as the encoding of the relationships between objects, actions, and temporal effects. Kjellstrom et al.[18] simplified the model to encode human-object interactions.

Using their model, it is possible to classify simple human-object activities by detecting hand-object affordances. Instead of having to build a model based partly on human poses or space-time activity volume like in classic approaches [19], this method allows us to model each activity as users interact with their environment. This approach also combines object detection, object tracking, and activity detection problems, since the person being observed can be represented as an object to track. This further simplifies the modeling of an activity as the tracking of object-action interactions [18]. This approach can extend naturally to simple temporal goal recognition [20] and can be used to extend activity recognition to activity prediction [21].

Assistive systems must also take into account the probability of erroneous activity classification, such as missing action detections, visual occlusions, and noise or anomalies in the activity execution. Fahad et al. [22] propose an algorithm to assign confidence scores based on every detected and missed action during activity execution based on their missing event rates and the actions' temporal events. Unfortunately, this approach cannot offer online erroneous classification.

B. Plan recognition

Plan recognition is the opposite of planning [3]. Similar to how high-level activity prediction aims to predict activities based on incomplete data, plan recognition aims to deduce the goals of the observed agent based on its perceived actions.

Plan recognition techniques often uses plan libraries, descriptions of plans as collections of observations related to the plan execution [5]. Plan libraries are a collection of symbols that represent goals, sub-goals and actions, and rules that represent logical links between these symbols. One of the formalisms of plan libraries is Plan Tree Grammar [5]. Plan Tree Grammar represents a plan as a hierarchical partially ordered AND/OR tree. In these trees, the roots correspond to the goal, and the leaves to the low-level actions. Its structure works well to express multiple interleaving goals, hierarchy in goals and actions, and observation loops. It also handles multiple constraint definitions in between actions.

The inference engine used for plan recognition over plan tree grammars consists of weighted Bayesian reasoning on a set of valid hypotheses. Generating this set is usually done with a top-down approach and is resource intensive, since generating all possible plans from the start goal and then pruning improbable and impossible plans provokes a combinatorial explosion. One solution consists in using a bottom-up approach on the tree by navigating from the leaves to the roots based on hypotheses generated by the observations. The bottom-up approach is computationally more expensive for each branch, but has less hypotheses to generate. Several projects use this approach [23] [6] but have had issues with execution time and noise robustness. This approach also loses its advantage with non-perfect observability, since it has to generate invalid hypotheses that can be skipped top-down.

Several studies in the domain propose using a decision model that corresponds to the preference for agents for certain goal execution [6] [9]. For example, certain ADL, such as "making coffee", can be performed in multiple ways, but each person has their own way of doing it that they usually follow. Mirsky and Gal [6] describe the decision model as a probability repartition over production rules.

In recent work, Massardi et al. proposed to use particle filters with a top-down approach to deal with both noisy observations and computation time [9]. It is an online approach that gives an estimation of the probability distribution over possible goals rather than an exact result of these probabilities.

C. Combined approach

There has been recent interest in the goal of using both activity and plan recognition in the field of daily living activities. Rafferty and Nugent [24] use a sensor-based approach to model possible activity beliefs based on the triggered sensors and an ontological rule-based goal recognition system. They manage to achieve perfect recognition accuracy in a controlled environment without noise, but their accuracy decreases to 60% when used in a real situation. This diminution can be attributed to the context information lost by their sensors, which can introduce noise in the activity recognition.

Granada and Pereira [25] proposed using video frames from training videos to directly train two convolutional neural networks (CNNs) on activity and plan recognition and got a maximum 67% recognition rate. Their precision rate decreased in many activities from using non-flexible and overly generic features when training their models as they did not train them on specific features, but on the training videos themselves.

While these techniques offer perspective on combined approaches using both low-level activity and plan recognition, there is a gap between state-of-the-art plan recognition techniques and what is applied in ADL recognition. One of the issues is that ADLs are usually seen as single isolated actions, such as *stand up*, *read*, *make a phone call*, rather than groups of actions with hidden goals, like *make a cup of tea*, which require several lower-level actions, such as *get cup*, *heat water* or *choose tea*.

III. PARC Architecture

PARC, as shown in Figure 1, works as follows. The robot uses its RGB-D sensor to capture color and depth data from the environment, which is sent to the activity affordance recognition component for treatment. The module then extracts the current activity being performed by the person being analyzed by PARC, if there are any.

To do so, the module locates and creates features from the user and objects present in the scene using its object model implemented by a region-based convolution neural network that uses YOLO V3 architecture [26]. The objects and human features are then analyzed to locate possible human-object affordances. They are then turned into possible activities based on their probability, the previous activities, and the activity beliefs of the plan model. The activity with the highest probability then goes to the plan recognition module to update the current plan model and its generators. The module then finally sends the current plan and future activity beliefs to the robotic agent.

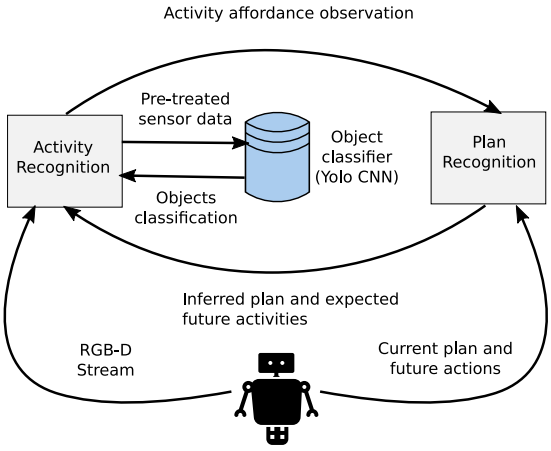

Fig. 1: PARC architecture

A. Activity-Affordance Recognition

This module receives color and depth fed from the sensor. The depth data, D, which is turned into gray-level intensity values for treatment, where each depth pixel can be transformed into distance values in meters.

1) Affordance features extraction: The color sensor data is sent to a region-based CNN[27] either using modern CPU optimization techniques [28] or using a GPU to perform objects and hands classification. The system outputs the vector Cl_t^o, where each element is composed of features of the located classes : o_t, r_t, p_t where o_t is the object class, r_t is the object bounding box, and p_t is the object probability. We then augment them by computing depth features from the depth grid we created earlier. To get a more precise depth from our non-segmented regions, we use a clustering algorithm, as described in Algorithm 1, to remove aberrant cell depth, as shown in Figure 2. The algorithm separates the depth data from the bounding box: D into n distinct regions c. It then compares the depth of that region and that of the one at the center of the bounding box, c_mil, and checks whether the depth data differs too much between the two. If the depth data is too dissimilar, the region is removed and the general depth data of the object is recalculated.

With the cells holding depth data too dissimilar to the main object and probable affordance location removed, the component computes the average depth of the object d_t and adds it to the features set. The features set is then split based on a body parts-objects dichotomy to generate two distinct sets, H_t,O_t. The resulting features are then sent to the Activity-Affordance recognition module.

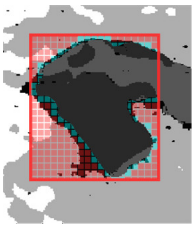

Fig. 2: Depth cells clustering. The depths corresponding to the object region are clustered together, removing most of the regions with aberrant depth values, as shown in the figure in red. The blue zones represent the acceptable depth error rate.

Algorithm 1 Depth mean clustering algorithm

for all $i \in Cl_t^o$ **do**
  $C \leftarrow \{c_1, c_2 \ldots_n\} \in r_i \subseteq D$
  $c_{mil} \leftarrow$ cell in position $\{len(c)/2, len(c)/2\}$
  $C \leftarrow C \setminus c_{mil}$
  for all $x \in C$ **do**
    if $d(depth(x), depth(c_{mil})) > 0.5$ **then**
    ▷ d(X,Y) calculates the distance on the depth axis between the two positions X and Y.
    ▷ If the depth differ too much, we remove the cell.
      $C \leftarrow C \setminus x$
    end if
  end for
end for

2) Activity-Affordance recognition: Similar to [29], time is discretized to the frames of the video feed, segmenting them into t temporal segments. Each segment corresponds to the timespan of frames passed between detections of new human-object affordance. Every frame, the system receives the features set $\{\omega_H, \omega_O\}$, where ω_H and ω_O are the current observations of the user's hands and objects. Each of those are added to their respective sets H^t, O^t, which describe the complete observations made on the person and objects in the environment since the last perceived activity. Those values are passed through the equation:

$$P(OA^t|H^t, O^t) = \prod_{i \in H} \prod_{j \in O} (dist(i,j) distDepth(i,j)) \quad (1)$$

which returns the probability of an affordance between human and object based on the features extracted in the

previous section. Since the system considers each activity as a human-object affordance, the predicted affordances are transformed into predicted activities, named A^t. The system then calculates the most probable predicted activity based on the previous probabilities and future actions predictions previously returned by the plan recognition module.

$$max\{P(A^t|\omega_{OA}) = P(A^t|OA^t)\sum_{x=i}^{t-1} P(OA^t|OA^x, H^t, O^t)\} \quad (2)$$

This value is finally sent into our goal recognition module as the current observed activity.

B. Goal Recognition

We use the goal recognition approach proposed by Massardi et al. [9]. This approach has two advantages for our application: (1) it is online, which means it can be used in real-life settings with time constraints; and (2) it can handle noisy observations. The following section will also provide a quick introduction to critical concepts as well as a presentation of plan recognition using particle filters.

As stated in Section II-B, in a typical goal recognition problem, the observations correspond to low-level actions performed by the observed agent, which, in this case, are the affordance activities sent by the activity recognition component. In our case, goal recognition also has to be done under real-time constraints and under noisy observation. Generally speaking, noise in goal recognition can be separated in three types: (1) missing observations, where the observer misses actions done by the observed agent; (2) mislabeled observations, where an observation is mixed with another one; and (3) extraneous actions, where actions which are not part of the plan are observed [3].

We rely on a library-based approach, as described in Section II. Our approach uses a description of the plan library as a context-free grammar. We describe a plan library as defined by Geib and Goldman [5].

Definition 1. *A plan library (PL) is a tuple* $PL = (A, NT, G, R)$ *where :*

- A *is a finite set of terminal symbols;*
- NT *is a finite set of non-terminal symbols;*
- $G \subset NT$ *is the set of goals;*
- R *is a set of production rules in the form* $\alpha \rightarrow S, \sigma$*, with* $\alpha \in NT$*,* S *a set of symbols from* $A \cup NT$ *and* σ *a partial order of* S*.*

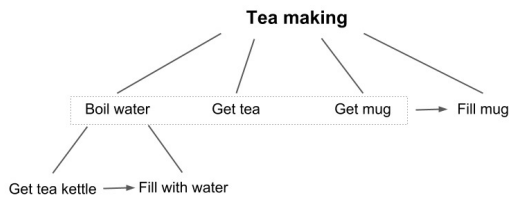

Fig. 3: Example of a small plan tree

Figure 3 shows an example for a single root goal and describes the actions needed to make a cup of tea. In this example, the terminal actions $Get\ tea$, $Get\ mug$ and the non-terminal action $Boil\ water$ can be done in any order, but they all have to be done before doing $Fill\ mug$. $Boil\ water$ can also be broken down into terminal actions.

To solve the goal recognition problem, we use the top-down error-tolerant anytime approach using a particle filter [9]. Compared to a normal top-down approach that generates all plan hypotheses and then prunes them based on the observations, this approach expands one hypothesis at a time. When expanding, the hypothesis is checked to ensure it is still valid given current observations. If not, it is dropped to limit the total tree size. We also use the eq.3 when expanding the tree to include the noise directly in the algorithm, where C is the list of children, $RNC(X, Y)$, random choice, is a algorithm that chooses randomly from the list Y based on the probability distribution X and where $child$ is the current tree node.

$$C(child) \leftarrow < child, \emptyset, RNC(DM, child), False > \quad (3)$$

To do so, the algorithm needs a noise model which models the prior probability of the different kinds of noises for each possible observation.

We use this approach to generate a population of plan trees to use them as particles for our particle filter algorithm. Rather than developing the complete plan, we delay the commitment by expanding one low-level action at a time. We randomly choose whether this low-level action will be expressed with noise using a probability distribution of noisy scenarios. If noise applies, we express it with three different cases: (1) mislabeled actions, we randomly choose another terminal symbol; (2) missing observation, we expand the particle to obtain the next low-level action; and (3) extraneous actions, same as mislabeled, but we keep in memory the actual symbol for the next particle expansion. This output symbol is called the last emitted symbol.

The particle filter algorithm can be summarized as follows:

1) Create a population of particles (plan trees) and generate the first emitted symbol for all of them using a top-down approach.
2) Sort them by the last emitted symbol.
3) On a new observation received from the activity recognition component, keep the particles where the last emitted symbol correspond to the observation.
4) Increase the population by copying some of the remaining particles.
5) Generate the next emitted symbol for all of them and go back to Step 2.

To solve the goal recognition problem, we just have to count the number of corresponding particles in the population map for each possible goal, the most probable goal being the one with the most particles. This goal is then transmitted back to the robot and the three most probable future activities are sent back to the activity recognition module.

IV. PARC IMPLEMENTATION AND EXPERIMENTATION

Since the goal of PARC is implementing activity and plan recognition for low-cost robotic assistants, the robotic platform is designed to limit the number of sensors and moving parts needed while still achieving optimal results.

Fig. 4: PARC robot prototype

Figure 4 shows PARC, our current robot, analyzing the current activity of an off-screen user. In terms of hardware, PARC is equipped with an off-board ASUS Zenbook with an Intel Core i5-6200U CPU, 8GB RAM and no GPU, an Intel RealSense D-435 RGB-D camera, a tripod and a Turtlebot 2e mobile platform[1]. On the software side, PARC features our activity and plan recognition components, both implemented in C++, alongside off-the-shelf OpenCV libraries for image treatment and deep learning in a CPU only environment, using our custom CNN architecture based on YOLO V3 [26], modified to work in a CPU and GPU environment with less layers.

A. Dataset

We did not find any public datasets using both video input and structured complex activities which would require plan recognition. Most of the video datasets for ADL rely on single activity approaches [15][21]. The closest we found were the Kitchen Scene context-based gesture recognition dataset [30] and the high-level activity dataset CAD120 [31], but they: (1) are not structured as plan recognition problems; (2) rely on context; and (3) rely on temporal actions. PARC can not currently handle point (2) and (3) and extensive work would be needed to adapt these datasets and would not show the integration of plan recognition for ADl recognition.

We propose two datasets: a small ADL dataset which is used as a proof of concept, and a direct implementation of plan recognition problems with a video support.

B. Experiments

1) Activity and plan recognition over a small domain: As stated in Section IV-A, a new dataset was created to test the approach since modern activity and plan recognition datasets either opt to modelize individual actions with no long-term planning or use non-visual sensor data. Our new dataset is composed of diverse typical possible ADLs that our system can find in everyday living. Compared to most activity prediction datasets which recognize human-only actions, human-robot interactions or action prediction [13] [14] [15] , and plan recognition datasets, which recognize non-visual actions for their dataset [32], our dataset is oriented to provide plans with similar activities in which one system needs efficient future activity prediction to recognize the current plans and actions.

Our dataset, named Teadomain, is composed of the following plans, actions and classes: {Tea-making, Chocomaking, Coffee-making}, {Hold}, {Teakettle, Teapot, Teabag, Mug, Milk, Coffee, Coffeemaker, Chocolate, Water pitcher, Can}

To build the dataset, we invited five people aged 20 to 30 to perform two to three different plans in front of the system, in variant settings. Each plan could be followed by performing a different set of sub-activities. Here are examples of some of the plans:

- Chocomaking(<Hold(teakettle), Hold(Waterpitcher)>,Hold(Mug), Hold(Choco))
- Chocomaking(Hold(Mug),Hold(Milk), Hold(Choco))
- CoffeeMaking(Hold(coffee),Hold(coffeemaker), Hold(water),Hold(coffeemaker),Hold(mug), Hold(milk),Hold(mug))

Each plan represents a set of sub-activities linked to the plan. Each activity must be performed in order, except for those inside the markers <>, meaning that those activities must be predicted at this stage in plan execution, regardless of their order.

For each plan, we recorded color and depth data at a resolution of 1280x720 pixels with a frame rate of 30 fps. For half of the videos, the robot moved horizontally and vertically to introduce egomotion in the visual feed [15]. This added motion is essential to confirm the approach efficiency when the visual feed loses pertinent data while in movement due to motion blur, such as when the robot must place itself in strategic locations to see what the person is doing or when it is in the person's path and must step out of the way.

For the experiments, we used a cross-subject, cross-motion test. To do so, we split the dataset into four different categories, with the first two sets composed of Subject 1-2 plan execution with and without motion, and the last two sets composed of Subject 3-4 plan execution with and without motion.

Categories 1 and 3 are used for training our object classification CNN and the remaining categories for testing. The results are shown as confusion Matrix in Figures 5 and 6.

TABLE I: Additive CPU and memory usage

Type	CPU usage	Mem usage	Mean execution time
Object classification	**83.7%**	**4.9%**	**0.640**
Activity recognition	**0.01%**	**0.3%**	**0.03**
Plan recognition	**0.08%**	**0.1%**	**0.025**

On average, PARC successfully manages to perceive and predict the high-level plan in 80% of the test cases. The correctness of the results fell significantly in our three tests where the environment was never used for the training

[1]https://www.turtlebot.com/turtlebot2/

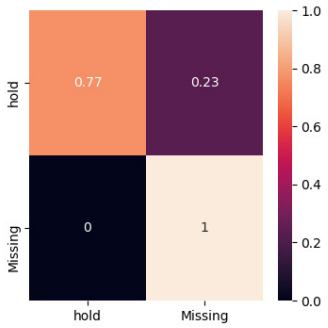

Fig. 5: Activity recognition confusion matrix. Missing label the number of possible activities missed by the system

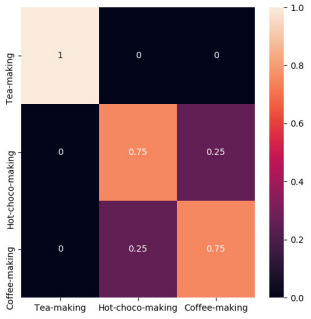

Fig. 6: Plan recognition confusion matrix

videos, which in turn made the object classification rate plummet due to variation in color, scaling, etc. This loss in object recognition seems to be the main cause for the loss in the activity and plan recognition rates. The system also has difficulty with plan sequences using the Coffeemaker and Milk, which had the lowest number of training pictures when training our CNN. This suggests that if the system training videos contained more sequences of activities then the recognition rate would increase.

Table I shows that PARC's only resource bottleneck comes from its object classification CNN model, and the rest of the system offers a high recognition rate with limited resource usage and low execution time.

2) Integration of plan recognition: For the second experiment, we focused on the integration of plan recognition with activity recognition. We used a simulated plan library, which is similar to the one used to asses the performances of several plan library systems [7] [6] [9]. This plan library consists of five low-level actions, five goals, a depth of 4, a branching factor of 3 for AND rules, a branching factor of 2 for OR rules and a partial order at 33%. The resulting plans are nine actions long.

We represented low-level affordance using colored balls. For example, to perform action a, a person had to take a blue ball and put it back. We took 50 videos of plan executions in different settings for our evaluation set and trained the system to recognize ball colours on 22 image sets with random interactions. Recorded values of color and depth are similar to those presented in the previous experiment.

Since the particle filter approach of plan recognition requires a noise function, we decided to use an over-estimation of the noise at 40%, with 30% of mislabeled actions and 10% of missing/extraneous actions.

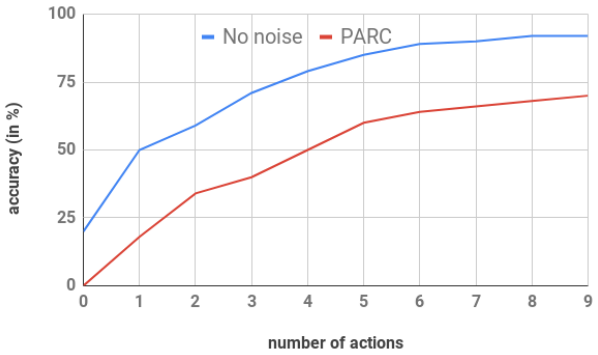

Fig. 7: Average rank of the correct goal by scenario completion

Figure 7 presents the accuracy of goal recognition by scenario completion for our videos. The blue curve represents the accuracy for perfect observability, i.e. plan recognition alone, and the red one represents our results with PARC.

We can see a shift in accuracy between observation 0 and observation 1 for the results. This can be explained by a difference between the plan recognition system, which tries to infer the goal before the first observation, and PARC, which tries to infer the goal only after the first observation.

PARC obtains a final accuracy of 70% on our dataset. This accuracy is slightly above what we expected using our noise model. We expected a value of 65%, but the difference is within acceptable margins. This experiment shows that PARC successfully integrated plan recognition and activity recognition.

V. CONCLUSIONS AND FUTURE WORK

This paper introduces PARC, a plan and activity recognition component designed to work with low-cost robotic ADL assistants. PARC uses an activity recognition module based on human-object affordance detection that is passed as observations to a novel plan recognition module based on particle filters to predict the current high-level plan of a person and their future actions. Our components helps the robot to understand a person's actions and infer the plan he is currently doing in order to provide specific aid to execute it. To achieve that, PARC successfully integrated activity recognition and advanced plan library based plan recognition in real-time/real-life settings.

In future work, we would like to focus on the addition of temporal values for low-level activities in the plan recognition system. Knowing when the current and future actions should end is needed for tasks like active plan recognition or intent recognition and can help PARC to understand high-level plans more clearly.

REFERENCES

[1] M. E. Pollack, L. Brown, D. Colbry, C. Orosz, B. Peintner, S. Ramakrishnan, S. Engberg, J. T. Matthews, J. Dunbar-Jacob, C. E. McCarthy, *et al.*, "Pearl: A mobile robotic assistant for the elderly," in *Association for the Advancement of Artificial Intelligence (AAAI) workshop on automation as eldercare*, vol. 2002, 2002, pp. 85–91.

[2] B. Graf, C. Parlitz, and M. Hägele, "Robotic home assistant care-o-bot® 3 product vision and innovation platform," in *International Conference on Human-Computer Interaction (HCI)*. Springer, 2009, pp. 312–320.

[3] G. Sukthankar, C. Geib, H. H. Bui, D. Pynadath, and R. P. Goldman, *Plan, activity, and intent recognition: Theory and practice*. Newnes, 2014.

[4] F. Amirabdollahian, R. op den Akker, S. Bedaf, R. Bormann, H. Draper, V. Evers, G. J. Gelderblom, C. G. Ruiz, D. Hewson, N. Hu, *et al.*, "Accompany: Acceptable robotics companions for ageing yearsmultidimensional aspects of human-system interactions," in *International Conference on Human System Interaction (HSI)*. IEEE, 2013, pp. 570–577.

[5] C. W. Geib and R. P. Goldman, "A probabilistic plan recognition algorithm based on plan tree grammars," *Artificial Intelligence*, vol. 173, no. 11, pp. 1101 – 1132, 2009. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0004370209000459

[6] R. Mirsky *et al.*, "Slim: Semi-lazy inference mechanism for plan recognition," *International Joint Conferences on Artificial Intelligence (IJCAI)*, 2017.

[7] F. Kabanza, J. Filion, A. R. Benaskeur, and H. Irandoust, "Controlling the hypothesis space in probabilistic plan recognition." in *International Joint Conference on Artifical Intelligence (IJCAI)*, 2013, pp. 2306–2312.

[8] S. S. Vattam and D. W. Aha, "Case-based plan recognition under imperfect observability," in *International Conference on Case-Based Reasoning (ICCBR)*. Springer, 2015, pp. 381–395.

[9] J. Massardi, M. Gravel, and E. Beaudry, "Error-tolerant anytime approach to plan recognition using a particle filter," in *International Conference on Automated Planning and Scheduling (ICAPS)*, vol. 29, no. 1, 2019, pp. 284–291.

[10] R. Poppe, "A survey on vision-based human action recognition," *Image and vision computing*, vol. 28, no. 6, pp. 976–990, 2010.

[11] L. Chen, J. Hoey, C. D. Nugent, D. J. Cook, and Z. Yu, "Sensor-based activity recognition," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 42, no. 6, pp. 790–808, 2012.

[12] D. Fortin-Simard, J.-S. Bilodeau, K. Bouchard, S. Gaboury, B. Bouchard, and A. Bouzouane, "Exploiting passive rfid technology for activity recognition in smart homes." *IEEE Intelligent Systems*, vol. 30, no. 4, pp. 7–15, 2015.

[13] A. Jalal, S. Kamal, and D. Kim, "Shape and motion features approach for activity tracking and recognition from kinect video camera," in *International Conference on Advanced Information Networking and Applications Workshops (WAINA)*. IEEE, 2015, pp. 445–450.

[14] J.-F. Hu, W.-S. Zheng, J. Lai, and J. Zhang, "Jointly learning heterogeneous features for rgb-d activity recognition," in *Conference on Computer Vision and Pattern Recognition (CPVPR)*, 2015, pp. 5344–5352.

[15] L. Xia, I. Gori, J. K. Aggarwal, and M. S. Ryoo, "Robot-centric activity recognition from first-person rgb-d videos," in *Winter Conference on Applications of Computer Vision (WACV)*. IEEE, 2015, pp. 357–364.

[16] J. J. Gibson, *The ecological approach to visual perception: classic edition*. Psychology Press, 2014.

[17] L. Montesano, M. Lopes, A. Bernardino, and J. Santos-Victor, "Learning object affordances: from sensory–motor coordination to imitation," *IEEE Transactions on Robotics*, vol. 24, no. 1, pp. 15–26, 2008.

[18] H. Kjellström, J. Romero, and D. Kragić, "Visual object-action recognition: Inferring object affordances from human demonstration," *Computer Vision and Image Understanding*, vol. 115, no. 1, pp. 81–90, 2011.

[19] J. K. Aggarwal and M. S. Ryoo, "Human activity analysis: A review," *ACM Computing Surveys (CSUR)*, vol. 43, no. 3, p. 16, 2011.

[20] D. Song, N. Kyriazis, I. Oikonomidis, C. Papazov, A. A. Argyros, D. Burschka, and D. Kragic, "Predicting human intention in visual observations of hand/object interactions," in *International Conference on Robotics and Automation (ICRA)*, 2013.

[21] V. Dutta and T. Zielinska, "Predicting human actions taking into account object affordances," *Journal of Intelligent & Robotic Systems*, pp. 1–17, 2018.

[22] L. G. Fahad and M. Rajarajan, "Anomalies detection in smart-home activities," in *International Conference on Machine Learning and Applications (ICMLA)*. IEEE, 2015, pp. 419–422.

[23] C. W. Geib, "Delaying commitment in plan recognition using combinatory categorial grammars." in *International Joint Conferences on Artificial Intelligence (IJCAI)*, 2009, pp. 1702–1707.

[24] J. Rafferty, C. D. Nugent, J. Liu, and L. Chen, "From activity recognition to intention recognition for assisted living within smart homes," *IEEE Transactions on Human-Machine Systems*, vol. 47, no. 3, pp. 368–379, 2017.

[25] R. Granada, R. F. Pereira, J. Monteiro, R. Barros, D. Ruiz, and F. Meneguzzi, "Hybrid activity and plan recognition for video streams," in *The AAAI 2017 Workshop on Plan, Activity, and Intent Recognition*, 2017.

[26] J. Redmon and A. Farhadi, "Yolov3: An incremental improvement," *arXiv preprint arXiv:1804.02767*, 2018.

[27] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *Advances in neural information processing systems*, 2015, pp. 91–99.

[28] S. Rajbhandari, Y. He, O. Ruwase, M. Carbin, and T. Chilimbi, "Optimizing cnns on multicores for scalability, performance and goodput," *ACM SIGOPS Operating Systems Review*, vol. 51, no. 2, pp. 267–280, 2017.

[29] H. S. Koppula and A. Saxena, "Anticipating human activities using object affordances for reactive robotic response," *IEEE transactions on pattern analysis and machine intelligence*, vol. 38, no. 1, pp. 14–29, 2016.

[30] A. Shimada, K. Kondo, D. Deguchi, G. Morin, and H. Stern, "Kitchen scene context based gesture recognition: A contest in icpr2012," in *Advances in depth image analysis and applications*. Springer, 2013, pp. 168–185.

[31] J. Sung, C. Ponce, B. Selman, and A. Saxena, "Unstructured human activity detection from rgbd images," in *International Conference on Robotics and Automation (ICRA),*. IEEE, 2012, pp. 842–849.

[32] N. Blaylock and J. Allen, "Generating artificial corpora for plan recognition," in *International Conference on User Modeling*. Springer, 2005, pp. 179–188.