RFID based activities of daily living recognition

Frédéric Bergeron and Sylvain Giroux
Laboratoire DOMUS
Université de Sherbrooke
{frederic.bergeron2, sylvain.giroux}@usherbrooke.ca

Kévin Bouchard and Sébastien Gaboury
Laboratoire LIARA
Université du Québec à Chicoutimi
{kevin.bouchard, sebastien.gaboury}@uqac.ca

***Abstract*—In this paper, we address the issue of indoor activities of daily living recognition with a novel Activity Recognition System (ARS). This system only uses interactions between objects. Their locations is provided by a tracking system based on passive RFID tags to compute activity probabilities. Classification within the tracking system is done through a random forest that gives over 97% in accuracy. Activities are represented by the use of a custom behaviour tree, which makes it possible to compose interactions to form activities of any complexity. Interactions and activities are defined in a human readable form to allow everyone to expand them with minimal prior knowledge.**

I. Introduction

Recent years have seen a need to perform certain tasks in a non-intrusive way to increase their acceptance by the targeted market. One of those tasks is activity recognition in a smart environment. Non-intrusive human activity recognition (HAR) is a difficult task with many modern applications. It can be exploited to detect dangerous or unusual situation given certain constraints. For instance, it can detect abnormal behaviour in an airport for security measure. The application targeted, in this paper, is activity monitoring for smart homes, which is a particularly challenging instance of this task.

This work is motivated by the urgent need for solutions concerning the rapidly growing proportion of the elders in occidental societies [1]. Activity recognition is an important technique needed to provide assistance to elders in smart homes to reduce resources demands to healthcare systems. Some applications in this context include cooking monitoring, autonomy evaluation, and simple general assistance. It can even be used to gather data for social studies on many topics, for instance on habits.

A smart home is a regular housing unit that has been enhanced with ways to collect data from what is going on inside and sometimes with direct ways to act on its environment or to interact with its inhabitant. There are many sources of data in a smart home. The DOMUS Laboratory embed a full smart home infrastructure containing more than two hundred sensors including twenty RFID readers, several motion detectors, infrared sensors and electric usage sensors. Fig.1 is a picture of the DOMUS smart home.

Tracking and positioning inside a smart home can be achieved using multiple technologies with varying precision and accuracy. Recent researches focus mainly on Bluetooth, Wi-Fi, UWB and RFID [2]–[8]. Ultrasound, microphones and very low resolution camera can also be exploited. There is also the well-known GPS technology one can use, even if it loose in precision in an indoor environment. The precision with these technologies varies from a couple meters to about 15 centimeters. However, they are not well adapted to track day to day objects (e.g.: a coffee mug or a plate).

Fig. 1. A picture of the DOMUS smart home

This paper presents an activity of daily living recognition system (ARS) for smart homes. Even though this system aims to provide assistance to the elders, the ideas presented can be applied to any application. The ARS focuses only on objects tracking data generated by a RFID positioning system developed previously at the DOMUS laboratory. While there could be many other sources of relevant data for the activity recognition task, using only one makes it easier and cheaper to deploy the system in another house. This ARS also has the particularity to be working in real-time, meaning it detects activities within a small fixed delay after it happened, 100 milliseconds in our experiments. It contributes to the domain by:

- proposing a flexible representation of activities using behaviour trees;
- introducing an efficient human activity recognition system that can resolve object equivalence. The system is also tolerant to noisy data.

II. Related work

The ARS presented in this paper uses data provided by a tracking system built on RFID readings [9], [10]. There are two families of RFID sensors, active and passive. Active tags contain an internal power source that passive tags do not have. Instead, they emit by recycling the energy contained within an inbound signal from the emitter. This inner power source allows for a stronger signal and more precise methods. For instance, Cory Hekimian-Williams and his colleagues implemented phase differences to achieve millimeter accuracy in perfect conditions for tracking [11]. Nevertheless, passive approaches give interesting results by using different methods. Thus, Fortin-Simard et al. [12] worked on a trilateration system. They exploited multiple filters to preprocess the received signal strength from passive tags as a preliminary step to an elliptical trilateration. Their method achieved a localization precision of up to 15 cm. What passive RFID loses in precision, it gains in other factors considered interesting for indoor HAR. They are much cheaper than active tags. They are also smaller and less cumbersome as they do not need to carry an energy supply.

Activity modelling is another popular topic among the HAR community. In [13], Fortin-Simard presents a HAR system that uses rules to detect activities. They start recognizing an activity once start rules are validated and end conditions are no longer detected. Palmes & al. [14] analyzed activities and concluded that the list of relevant objects we use for an activity are very similar no matter how the activity is performed. They, therefore, compute the relative weight of objects within an activity using data mining techniques on recipes found on the internet. Those weight are used to find the boundary between activities with the *MaxGain* and *MaxGap* algorithms they presented. Tao Gu & al. exploit Emerging Pattern to recognize both single-user and multi-users activities [15]. Emerging pattern is an algorithm that finds significant changes between classes of data to perform classification. Onthologies have also been used before to model activities and their context [16]. Rashidi developed an automated system to discover activities and monitor them using a sequence mining algorithm [17].

III. Indoor Tracking

The indoor tracking system (ITS) is the module on which all this work rely [9], [10]. It classifies the received signal strength indication (RSSi) from the twenty RFID antennas of the smart home to determine in what zone the tag is. The surface of the smart home was divided into sections, or zones, of various dimensions to obtain a high accuracy while also maintaining the precision needed for activity recognition. The zones in the kitchen are of a smaller dimension than the zones in the living room since activities performed in a kitchen usually imply movements of smaller range. They are 40cm X 40cm and zones in other rooms range from 60cm X 60cm to 75cm X 75cm. The classification itself is done by a single random forest of 250 random trees. In [10], it was mentioned that the apartment was divided into rooms for each of which a specific classifier was built for the zones it contained. The other sensors of the apartment were used to determine the current room the resident was in. However, the datasets were merged and the methodology improved to remove the need for this assumption. The accuracy of a random forest on this dataset is computed using 10-folds cross-validation within Weka [18]. With respectively 1, 10 and 50 or more three we obtain 88.73%, 96.24% and 97.2%. The unbalanced average accuracy for the previous room datasets was 92.2590% with 100 random trees, which is lower than the actual accuracy of about 97%. The gain comes from the modification of the zones layout (bigger than before). Also, since the RFID antennas are placed to cover all the surface of the apartment, only some antennas are relevant for each zone. For this reason, it makes no real difference for the random forest if we add more zones to classify since there are still approximately the same number of zones per antenna, and thus the same dimensionality within the data. Also, in this paper, we had to adapt our previous filters to the new tracking based on a single global dataset. To do so, we first added all the walls of the apartment to the knowledge base. Then, we made each zone responsible to know all its neighbour. This is required because zones can have different sizes and therefore many neighbours in a given direction. Finally, when deciding if a move is authorized or not, we now check if it implies crossing a wall in addition to looking for teleportation. The resulting movement is then chosen accordingly to the neighbours.

IV. Activity modelling

In the introduction, we mentioned that this ARS has only access to tracking data from objects. In fact, those tracking data can be seen as a triplet consisting of the object's name and type (like `cup-1`), its position (like `a2`) and the time (precise to milliseconds) at which this position was recorded. The position itself is a relative one, a *zone* that can have any dimension as explained in the previous section.

A. Interaction types

The ARS only has access to the positions of tagged objects. This means that this ARS cannot recognize some common activities like reading a book or watching TV where the object does not move. Another limitation comes from the fact that not all objects can be tagged. Comestible items that are contained in some sort of container, like coffee, can be tracked while free item like apples cannot. RFID tags are also not oven or microwave oven friendly as their components tends to fry in those conditions. On the other hand, many activities can be described as one or many interactions between objects. In our context, an interaction is a timed distance relation between two objects, a concept previously explored by Bouchard & al. [19]. We consider three types of relation:

- **SAME**: The two objects are in the same zone.
- **CLOSE**: This is when two objects are within *X* zones of each other. This also applies when the positions are the same. In fact, the exact distance *X* is related to zone dimensions. It can be expressed in a number of zones or in a real distance (like centimeters).
- **FAR**: This is when two objects that were at least close are separated.

To better explain the use of every type of relation, let us look at three concrete examples. The relation **SAME** can describe the interaction of pouring water into a glass. In this case, the water pitcher would be at the same position as the glass.

The relation **CLOSE** can apply to the interaction between chocolate dust and a mug. One could choose to pour chocolate dust directly from the container to the mug while someone else could choose to use a spoon, so the positions might be slightly different but still not too far away to avoid spilling the dust.

Finally, we can use the interaction of type **FAR** to model the action of undressing the table. In that case, a plate would move away from the table to the kitchen cabinet.

It is apparent that using only interaction between two objects is too restrictive. Indeed, many activities of daily living imply only one object. Taking out the trash is a perfect example of such activity. There is only a single object, the trash bin, and since we usually move it only to take it out, our ARS should be able to detect it. This led to the introduction of the *fake objects* concept. Fake objects are immutable point of interests added in the system to enable the use of discrete locations as objects. This way, a trash spot can be defined and the interaction type **FAR** can be used to detect the activity of taking out the trash. Another example of fake object is the kitchen sink to enable recognition of filling things with water.

B. Interaction composition

The interaction types allows recognition of atomic activities. However, many activities cannot be described using only a single interaction. Washing a glass, for instance, needs an interaction to add soap in water and another to put the glass in the water. In that case, some interactions might need to occur in a given order, or that one may be equivalent to another one and thus only one of them is required.

1) Behaviour tree: Those constraints can be expressed simply both for a human and a computer using a behaviour tree. In a behaviour tree, there are two different types of nodes to express different behaviour concerning their children nodes or leaves. The first node type is the *sequence*. This is used when the children must be evaluated in an ordered way, from left to right. This means the second child of a *sequence* node will only be considered once the first one is done. The second type is the *selector* node. It is used when there is no specific order for the children to be completed. However, evaluation is still made from left to right, so children nodes should be placed in order or realization probability.

The basic behaviour tree model also relies on a concept of state for the node and leaf. The state can be default, ongoing, failure or completed. In our context, it is more useful to have an order of completion since we are dealing with timed interaction. The state is then a percentage of completion, from 0 when the interaction has not yet started to 100 when the minimal time for the interaction has been reached. When the state is set to 100, it puts the node in a completed state where it stops evaluating its children for efficiency reasons.

The types of nodes are modified to fit our context. First, the *selector* node is replaced with a *parallel* node. The behaviour of the *parallel* node differs in the way that all children node have the same probability of happening. It means that all children node must reach 100 percent completion for this node to complete, but they can do so without any preferred order. A third modification is to add a third type of nodes to reflect the choice. It behaves like the *parallel* node with the simple difference that the node's completion is the highest completion percent among its children. We call it the *any* node. This way, all choices have the same probability of realization and the first one that completes is considered to be the chosen one.

The completion of each type of node can be summarized this way: for *sequence* and *parallel* nodes it is the sum on of the completion of each child divided by the number of childrens and for the *any* node it is the highest among the childrens. The leaves of the tree are the interactions between objects where the percentage of completion is directly proportional to how much time the objects stayed in the wanted relation compared to the minimal time they should spend. This ratio is expressed as a percentage. While these trees can be used to aggregate any interaction into an arbitrary complex activity, they can also be used to compose many activities together to form a more complex one. For instance, this allows to recognize and follow the progress of any cooking recipe by composing each step as a mixture of sub-activities and sub-interactions.

V. Activity recognition system

In the previous section, we described how we represent activities in term of a behaviour tree of interactions in our ARS. This section presents how we use those representations to actually recognize an ongoing activity in real time.

This first step to recognizing activities is to recognize interactions. Like we said, interaction occurs between two objects, real or fake. The difficulty resides in finding between which two objects. Indeed, there is usually more than one instance of each object. There is more than one cup we can use to make coffee, for instance. There are also many equivalent objects. While it may seem inappropriate, it is not impossible to take a glass instead of a cup to drink coffee.

A. Candidate interactions

To find between which two precise objects an interaction is occurring, every possible pair is considered. To do this, we introduce the concept of *candidate interaction*. There is a candidate interaction for each possible pair of objects that may be implied in a given interaction. Each time the position of an object is updated, all candidate interactions for this object are evaluated to check if the interaction type is respected. If it is the case, the time progress is updated according to the time associated with the position. The progress percentage of an interaction is highest among its candidates. There may be more than one candidate with progress since there may be more than one person moving things, or they may be noise in the data.

The complete mechanism of the evaluation on an interaction is better explained with an example. Let us take the interaction pouring water into a glass. We have three glasses, A, B and C. We have a sink in the kitchen, K, and another one in the bathroom, T. The candidate interactions are {A,K}, {B,K}, {C,K}, {A,T}, {B,T} and {C,T}. The target minimal time for the interaction is 5 seconds and the interaction type is **SAME**. To simplify, we consider that the positions are updated simultaneously each second. Therefore, a candidate interaction is completed if its two objects are in the same zone for 5 seconds. Table I gives the positions in the form of *(object, position)* and the states for updated interaction as $\{X_1, X_2\}$:progress percent.

This toy example shows well the whole process. When two objects begin interacting, a time counter starts. As long as they stay in place, the progress is noted. When one of them moves, the progress stops. In this example, the evaluation was tolerant to a one second noise. This is what happens for the interaction {A,K} at time 4. At the next second the glass comes back, so we consider it to be a temporary noise and we compute progress as if it did not happen. At time 4 the candidate {C,T} also sees its progress stopped by a shift in the position of C. However, since C does not return to the position of T, the whole progress is discarded.

B. Object equivalence

Another difficulty with activity recognition based on interactions between objects is to consider equivalent objects. A simple way to overcome this is to consider categories of objects instead of precise objects when modelling. For coffee, we would then use a *heat resistant recipient* and for cold water a *recipient*. In this example, *heat resistant recipient* is a subset of *recipient*, meaning that all objects that qualify for the first category also qualify for the second. This demands to create a clear hierarchy of objects that inherits properties from their superclass, much like the object model in object oriented programming. An appropriate way to represent categorization like this would be to interrogate an ontology. However, ontologies are not yet fast enough to respond to this type of queries in less than a firm twenty milliseconds. The solution we used was to manually build a tree representing the hierarchy and write the code to use it. This is a little more tedious to maintain but much faster given current ontology engines.

C. Activity manager

Now that we have a mechanism to compute interaction progress, we can explain how it is done at the activity level. The probability of an activity is related to the completion degree of each of its interactions when composed with the rules associated with each type of node presented in section 3. Much like an object could be part of more than one interaction, an interaction can also be part of many activities which can then be part of higher order activities and so on.

Those probabilities are collected by an activity manager whose is charged with doing the actual recognition. To do so, he normalizes the probabilities for all activities. He then confronts the most probable activity against a confidence threshold. If the probability is higher than the threshold, it is recorded as being formally recognized. Otherwise, it means that too many activities have a high probability and that the manager should wait a little more to make sure.

The main drawback of this method is that the manager will only recognize a composition of activities after its activities have all been recognized. A simple fix is to consider the composition of activities as a separate category and normalize them apart from the others. This way, we can detect higher order activity when enough sub-component have progressed.

The whole system has not been tested in production with complex activities yet. However, it has been tested with the simple activities presented as examples in this paper. Other tests also include specific scenarios like making instant coffee versus making tea. The ARS shows a perfect score on those scenarios with the exact time needed to recognize varying with the amount of noise received from the RFID modules giving the positions. Here is a selected list of activities that the ARS can recognize:{make coffee, make tea, empty trash bin, empty recycle bin, empty compost bin, drink water, make porridge, cook egg(s), vacuuming, drink milk, drink juice, use butter}

VI. Discussion

In the previous sections, we presented the activities of daily living recognition system we built using only the position of objects. Even though early results are promising, more tests with more complex activities ought to be completed.

Using objects interactions brings some limitations, as discussed before. It makes it hard or even impossible to identify activities that involve zero or one objects. Adding a tag to a person could help with some activities, but there is no guarantee the person would wear its tag at all time. Moreover, it could be considered as intrusive for some person to know that their position is continually tracked while at home. On the other hand, it would also enable to create a probability distribution for everyone in the smart home.

A strength of our approach is that it allows parallel activities to be performed without significantly affecting the recognition, given a high enough value for tolerance parameter in the interaction. It is also insensible to the multi-inhabitant problem if activities are not done at the exact same time. It is possible to ask the activity manager to perform many normalizations after removing the highest activity to find if the second one also meets the threshold.

Another interesting feature of our system arises when we consider the completion percentage as a score. This way, the interactions become a single source of score among any other sources of knowledge. We could, for instance, add another probability score based on the time of the day or the users' preferences. Moreover, it would allow for a certain probability modifier to increase the probabilities of certain activities by knowing that another one has been completed recently. For instance, it would allow faster recognition of the activity *drying the clothes* knowing that they have previously been washed.

VII. Conclusion

In conclusion, we presented an activity recognition system oriented towards activities of daily living recognition in a

TABLE I
EXAMPLE OF CANDIDATE INTERACTIONS

Time	Positions	Candidates progress	Remarks
0	(A, a1), (B, d2), (C, e4), (K, a3), (T, w2)	A,K:0, B,K:0, C,K:0, A,T:0, B,T:0, C,T:0	
1	(A, a3), (B, d2), (C, t4), (K, a3), (T, w2)	A,K:0, B,K:0, C,K:0, A,T:0, B,T:0, C,T:0	The candidate A,K begins to count time.
2	(A, a3), (B, d2), (C, w2), (K, a3), (T, w2)	A,K:20, B,K:0, C,K:0, A,T:0, B,T:0, C,T:0	The candidate while C,T begins to count.
3	(A, a3), (B, d2), (C, w2), (K, a3), (T, w2)	A,K:40, B,K:0, C,K:0, A,T:0, B,T:0, C,T:20	
4	(A, a2), (B, d2), (C, w1), (K, a3), (T, w2)	A,K:40, B,K:0, C,K:0, A,T:0, B,T:0, C,T:20	The candidates A,K and C,T pause.
5	(A, a3), (B, d2), (C, w1), (K, a3), (T, w2)	A,K:80, B,K:0, C,K:0, A,T:0, B,T:0, C,T:0	The candidate A,K resumes, C,T is discarded.
6	(A, a3), (B, d2), (C, w1), (K, a3), (T, w2)	A,K:100, B,K:0, C,K:0, A,T:0, B,T:0, C,T:0	The candidate A,K completes.

smart home. This ARS uses tracking position of objects provided by a random forest to detect interactions between objects or places and infer activity probabilities by composing them using a custom behaviour tree. The tracking positions come from RFID tags that are placed on objects, which prevents some categories of item from being tracked. The ARS allows recognition of activities of any complexity, as long as they can be described using interactions between objects or between a specific place and an object.

Activities implying objects at no specific location or no objects at all are detectable by our system. However, it is possible to combine it with other data source to extend our interaction based method.

REFERENCES

[1] W. H. Organization *et al.*, "World report on ageing and health," *Geneva: World Health Organization*, 2015.

[2] R. S. Ransing and M. Rajput, "Smart home for elderly care, based on wireless sensor network," in *Nascent Technologies in the Engineering Field (ICNTE), 2015 International Conference on*. IEEE, 2015, pp. 1–5.

[3] G. P. Joshi, S. Acharya, C.-S. Kim, B.-S. Kim, and S. W. Kim, "Smart solutions in elderly care facilities with rfid system and its integration with wireless sensor networks," *International Journal of Distributed Sensor Networks*, vol. 2014, 2014.

[4] J. Yim, C. Park, J. Joo, and S. Jeong, "Extended kalman filter for wireless lan based indoor positioning," *Decision support systems*, vol. 45, no. 4, pp. 960–971, 2008.

[5] J. Rossey, I. Moerman, P. Demeester, and J. Hoebeke, "Wi-fi helping out bluetooth smart for an improved home automation user experience," in *Communications and Vehicular Technologies (SCVT), 2016 Symposium on*. IEEE, 2016, pp. 1–6.

[6] L. M. Ni, Y. Liu, Y. C. Lau, and A. P. Patil, "Landmarc: indoor location sensing using active rfid," *Wireless networks*, vol. 10, no. 6, pp. 701–710, 2004.

[7] M. Altini, D. Brunelli, E. Farella, and L. Benini, "Bluetooth indoor localization with multiple neural networks," in *Wireless Pervasive Computing (ISWPC), 2010 5th IEEE International Symposium on*. IEEE, 2010, pp. 295–300.

[8] C. Yang and H.-R. Shao, "Wifi-based indoor positioning," *IEEE Communications Magazine*, vol. 53, no. 3, pp. 150–157, 2015.

[9] F. Bergeron, K. Bouchard, S. Gaboury, S. Giroux, and B. Bouchard, "Indoor positioning system for smart homes based on decision trees and passive rfid," in *Pacific-Asia Conference on Knowledge Discovery and Data Mining*. Springer International Publishing, 2016, pp. 42–53.

[10] F. Bergeron, K. Bouchard, S. Giroux, S. Gaboury, and B. Bouchard, "Qualitative tracking of objects in a smart home: A passive rfid approach based on decision trees," in *Proceedings of the 9th ACM International Conference on PErvasive Technologies Related to Assistive Environments*. ACM, 2016, p. 2.

[11] C. Hekimian-Williams, B. Grant, X. Liu, Z. Zhang, and P. Kumar, "Accurate localization of rfid tags using phase difference," in *RFID, 2010 IEEE International Conference on*. IEEE, 2010, pp. 89–96.

[12] K. Bouchard, D. Fortin-Simard, S. Gaboury, B. Bouchard, and A. Bouzouane, "Accurate trilateration for passive rfid localization in smart homes," *International Journal of Wireless Information Networks*, vol. 21, no. 1, pp. 32–47, 2013. [Online]. Available: http://dx.doi.org/10.1007/s10776-013-0234-4

[13] 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.

[14] P. Palmes, H. K. Pung, T. Gu, W. Xue, and S. Chen, "Object relevance weight pattern mining for activity recognition and segmentation," *Pervasive and Mobile Computing*, vol. 6, no. 1, pp. 43–57, 2010.

[15] T. Gu, L. Wang, Z. Wu, X. Tao, and J. Lu, "A pattern mining approach to sensor-based human activity recognition," *IEEE Transactions on Knowledge and Data Engineering*, vol. 23, no. 9, pp. 1359–1372, 2011.

[16] L. Chen, C. D. Nugent, and H. Wang, "A knowledge-driven approach to activity recognition in smart homes," *IEEE Transactions on Knowledge and Data Engineering*, vol. 24, no. 6, pp. 961–974, 2012.

[17] P. Rashidi and D. J. Cook, "Com: A method for mining and monitoring human activity patterns in home-based health monitoring systems," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 4, no. 4, p. 64, 2013.

[18] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, "The weka data mining software: an update," *ACM SIGKDD explorations newsletter*, vol. 11, no. 1, pp. 10–18, 2009.

[19] K. Bouchard, A. Bouzouane, and B. Bouchard, "Discovery of topological relations for spatial activity recognition," in *Computational Intelligence and Data Mining (CIDM), 2013 IEEE Symposium on*. IEEE, 2013, pp. 73–80.