*In Proceedings of IFIP 2007 International Conference on Ubiquitous and Intelligent Computing (UIC2007). Hong Kong, China. LNCS. July 2007. *best paper award**

Activity Recognition using an Egocentric Perspective of Everyday Objects

Dipak Surie[1], Thomas Pederson[1], Fabien Lagriffoul[1], Lars-Erik Janlert[1], and Daniel Sjölie[2]

[1]Department of Computing Science, Umeå University,
SE-901 87 Umeå, Sweden
{dipak, top, fabien, lej}@cs.umu.se
[2]VRlab / HPC2N, Umeå University,
SE-901 87 Umeå, Sweden
deepone@hpc2n.umu.se

Abstract. This paper presents an activity recognition approach based on the tracking of a specific human actor's current object manipulation actions, complemented by two kinds of situational information: 1) the set of objects that are visually observable (inside the „observable space") and 2) technically graspable (inside the „manipulable space"). This "egocentric" model is inspired by situated action theory and offers the advantage of not depending on technology for absolute positioning of neither the human nor the objects. Applied in an immersive Virtual Reality environment, the proposed activity recognition approach shows a recognition precision of 89% on the activity-level and 76% on the action-level among 10 everyday home activities.

1 Introduction

Inspired by "digital assistant" design research efforts such as [1], our vision is to design computer systems intended to facilitate personal, practical and user-defined everyday activities. Such systems should have awareness about the actor's current activity, situation and intention before the activity is actually completed to provide appropriate support. Capturing actor's intention implicitly is a very hard challenge, but could be simplified by capturing his/her situative information. This calls for an understanding of the actor's situative information while performing the activities from his/her perspective. Such systems can be used in providing assistance to people suffering early stages of dementia disease in completing their activities of daily living (ADL). Typical problems include the forgetting of performing an activity or an action within an activity; not being able to get started in the first place; not being able to continue after having been interrupted; or missing some operations that are mandatory for the completion of an activity. A system that could help overcome the above mentioned problems would enable patients to stay in their home for a longer period of time, have a normal independent life, and also reduce the burden on family members and caregivers.

Activity recognition based on user's body movements using accelerometers is described in [3], [4], and [5]. Such approaches are restricted to recognizing a subset of everyday human activities involving body movements like walking, running, brushing teeth, etc. Activity recognition based on audio is described in [6], but such approaches once again are restricted to a subset of sound-related activities. Recognizing activities using single but complex sensors like a camera is described in [7]. Such approaches are severely hampered by the complexity in extracting valuable features, even though human beings use the sense of vision effectively. Many activity recognition systems are heavily driven by currently available sensor technology. However, we believe that starting out from a perspective more centred around how humans literally perceive the world, based on the weight that current cognitive science give to this source of information as an activity-driving factor, could offer a valuable complement. In particular, it could offer a conceptual design platform robust enough to survive and handle generations of changes in the field of sensor technology.

Recognizing activities based on object manipulation is becoming an important research approach [10], [11]. We consider such an approach but extend it by also considering the changes to the set of objects seen and to the set of objects touchable by an actor in recognizing activities. According to situated action theory [12], the situation of an actor is a very important factor in determining what action the actor will perform. In performing various everyday activities, objects enter and leave the actor's observable space and manipulable space in a dynamic fashion. We recognize activities based on a simplified model of human perception limitations and considering Janlert's "proximity principle" [13], things that matter are close, and things that are close matter. Recognizing activities by attaching simple state change sensors to relevant objects (cf. [14]) is another interesting approach that we would consider in the future. Everyday human activities are modelled using a 3-level abstraction of activity, action and operation inspired from activity theory [2].

2 An Egocentric Perspective of Everyday Objects

The term 'egocentric' has been chosen to signal that it is the body and mind of a specific human actor that (sometimes literally, as will be shown later) serve as centre of reference to all his/her interaction with everyday objects.

2.1 A Situative Space Model: Bridging the Intent-Action Gap

A *situative space model* is developed on the basis of what a specific human actor can see and not see, touch and not touch at any given moment in time (Fig. 1). Human actors situate themselves closer to the objects relevant for their current activity. Such explicit situatedness gives an indication of the actor's intent (the needs and wants of the actor to satisfy some goal). By capturing the changes to the actor's observable space and manipulable space, we are indirectly capturing the actor's intentions.

The situative space model serves as a conceptual tool for determining what situative aspects of human activities need to be captured and in what detail. For our current prototype we consider only *the presence* of objects within the actor's observable space and manipulable space. In the future we intend to include *states* within individual objects to capture additional situative information. We also intend to include information about any other human actors present in the monitored spaces.

2.2 Applying the Egocentric Perspective

We have chosen to operationalize the situated space model (Fig. 1) as follows:

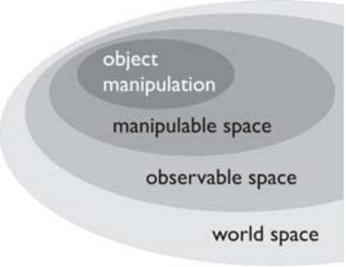


Fig. 1. A situative space model adapted from [15].

- The *world space* contains the set of all objects known to the system. In a real-world setting it might be determined by all objects carrying an RFID tag. In the Virtual Reality (VR) setting used for our evaluation purposes, it is the set of all objects included in the VR model simulating a kitchen environment.
- The *observable space* (OS) is the set of objects within a cone in front of the human actor's eyes with this cone following the head movements as shown in Fig. 2(a). The height of the cone is limited by the walls in an indoor environment and visual occlusion further affects the number of objects within this space.

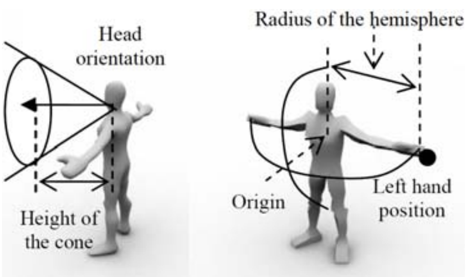


Fig. 2. **(a)** Observable space, **(b)** manipulable space.

- The *manipulable space* (MS) is the set of objects within a hemisphere in front of the human actor's chest (Fig. 2(b)). Such a shape is motivated by the fact that humans have two hands and the assumption that they manipulate objects within reach of their hands. The hemisphere follows the human actor's chest movements and its radius is equal to the maximum distance between the chest and a hand.
- *Object manipulation* (OM). When objects are manipulated by an actor, two events can be generated: *objectID_grabbed* and *objectID_released*. Although the actor can manipulate objects with both hands, we do at the moment not make any distinction between right and left. A similar approach is described in [16], [10] and [11]. Object manipulations represent the operations performed by an actor during the accomplishment of an action (and activity).

3 Activity Recognition System

Virtual Reality is used as a test-bed in order to speed up the design (and re-design) process and to compensate for the limitations with the currently available sensing technologies. A VR model, developed using the Colosseum3D real-time physics platform [17] is used to simulate a physical home environment with wearable sensors and sensors on everyday objects to capture an actor's object manipulation (OM), observable space (OS) and manipulable space (MS). Fig. 3 shows a snap-shot of the VR environment. OS and MS are captured at 1 Hz, while OM is captured when a grab/release event occurs. Refer to Fig. 4 for our system architecture. We have experimented with 70 object types. Object types include mobile object types like *fork*, *knife*, *plate* etc. and stationary object types like *microwave oven*, *sink*, *stove*, etc. We only consider the *type* of object in recognizing activities, not the identiy (e.g. *fork_1* and *fork_2* are both considered as *fork* type).

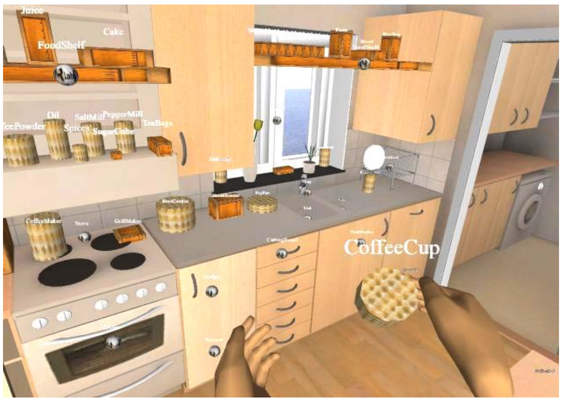


Fig. 3. Virtual Reality home environment.

There are many objects that overlap for several activities. For instance, *fork*, *knife*, *plate*, etc. are used for several activities like *preparing breakfast*, *preparing the table*, *having breakfast*, *doing the dishes*, etc. This makes the classification problem harder compared to taking an approach where the recognition system is strongly characterised by one or two objects that are unique to the activity. We do have some activities like for instance *preparing_rice*, where the *rice_bag* is a unique object for this activity. But this does not simplify the classification problem for the following reasons: 1) we are not only recognizing the actor's current activity, but also the actor's current action. The *rice_bag* is not manipulated by an actor while performing all the actions within this activity, but only while performing a few actions within this activity; 2) the *rice_bag* manipulation might be a noise created by the actor while performing another activity, and 3) the recognition system should recognize the activity and action before they are actually completed to provide appropriate assistance to the actor. Hence the system cannot wait until the unique object is manipulated to recognize the activity and action.

3.1 Feature Extraction

OS and MS both consist of sets of objects that need to be quantified. Our quantification scheme builds $\vec{S}_A$ and $\vec{S}_B$ as shown in Fig. 4, where the vectors represent the list of distinct object types with their corresponding number of occurrences. A log function is applied on $\vec{S}_A$ and $\vec{S}_B$ to give more importance to the type of objects present within those spaces compared to the number of their occurrences. One limitation of our quantification scheme may be its scalability to a large number of object types, since the dimension of the quantification vector depends on the total number of object types. We have experimented with 70 types of objects and have obtained good results.

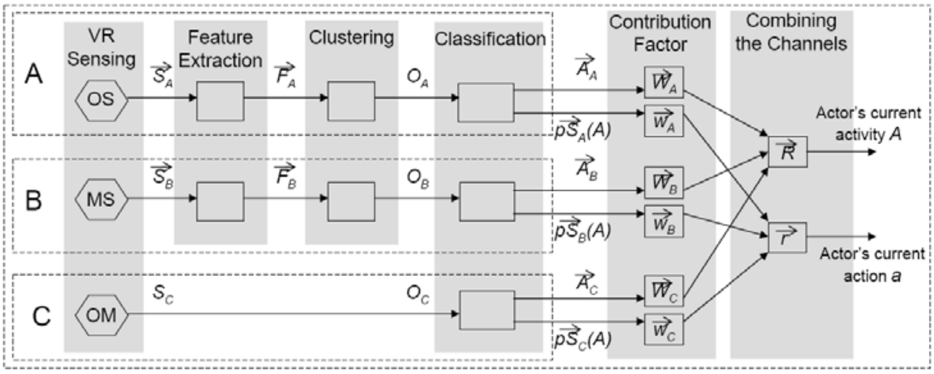


Fig. 4. The proposed activity recognition system architecture (D) combining three information channels (A, B and C) corresponding to the Observable Space, Manipulable Space, and Object Manipulation components pictured in Fig. 1 earlier.

3.2 Clustering

$\vec{F}_A$ and $\vec{F}_B$ are fed into two distinct clustering algorithms to retain the features provided within their respective information channels. We obtain the cluster centre for observable space (O_A) and manipulable space (O_B) which is used for further classification as shown in Fig. 4. Growing neural gas (GNG) clustering algorithm [18] was preferred compared to classical clustering techniques like K-Means or Kohonen SOM because it maintains a steady learning rate and also creates new clusters with additional training data. We do not want the end-user to fine-tune or parameterize the system for new situations added. However, we use the GNG algorithm as a classical clustering algorithm with fixed number of clusters for our current implementation. We intend to include the above mentioned features in the future.

3.3 Classification

The probabilistic generative framework of hidden-markov model (HMM) [19] is used because of its clear bayesian semantics, its ability to handle time-varying signals and the availability of efficient algorithms for state and parameter estimation. HMMs reduce the system's configuration space into a number of finite discrete states together with the probabilities for transition between the states. One limitation of HMMs is that the model structure has to be user-defined, which includes the number of states and the connections between the states. The model structure cannot be determined by standard learning methods. This should not pose a major problem since the activities recognized are user-defined. The actor provides ground truth for both activities and actions. Each activity is modelled using a separate HMM with the number of states corresponding to the number of actions within that activity. Similarly, the transitions between states correspond to the transitions between different actions within that activity. HMMs have shown good results in many activity recognition systems including [6], [8], [9] and [11].

The activity recognition system uses three information channels (refer to Fig. 4). Each information channel produces a sequence of observations that are fed into ten HMMs (one for each activity). For each information channel, the outputs from the ten HMMs are used to build an activity probability vector ($\vec{A}_A$, $\vec{A}_B$ and $\vec{A}_C$) containing the probabilities for each possible activity. The element of the activity probability vector with the highest value gives the actor's current activity and its most probable state gives the actor's current action.

3.4 Combining the Information Channels

The three information channels are combined using activity contribution factors ($\vec{W}_A$, $\vec{W}_B$ and $\vec{W}_C$) and action contribution factors ($\vec{w}_A$, $\vec{w}_B$ and $\vec{w}_C$). $\vec{W}_A$, $\vec{W}_B$ and $\vec{W}_C$ consist of the recognition precision values for each activity while $\vec{w}_A$, $\vec{w}_B$ and $\vec{w}_C$ consist of the recognition precision values for each action. These factors are automatically generated from the training data. We first determine the actor's current activity by computing $\vec{R}$, the weighted sum of all the three information channels:

$$\vec{R} = \vec{W}_A * \vec{A}_A + \vec{W}_B * \vec{A}_B + \vec{W}_C * \vec{A}_C \qquad \textbf{(1)}$$

Where * represents element-by-element multiplication. The element of $\vec{R}$ with the highest value gives the actor's current activity *A*. Once the activity is known, we determine the actor's current action by calculating $\vec{r}$ using the following formula:

$$\vec{r} = \vec{w}_A * p\vec{S}_A(A) + \vec{w}_B * p\vec{S}_B(A) + \vec{w}_C * p\vec{S}_C(A) \qquad \textbf{(2)}$$

Where $p\vec{S}_A(A)$, $p\vec{S}_B(A)$ and $p\vec{S}_C(A)$ are the vectors of state probabilities of the HMM representing the actor's current activity *A* for observable space, manipulable space and object manipulation respectively. $\vec{w}_A$, $\vec{w}_B$ or $\vec{w}_C$ is equal to zero if their respective channel is not supportive to the activity determined previously. The element of $\vec{r}$ with the highest value gives the actor's current action. We have not combined the three information channels before classification since we want to evaluate them independently, allowing us to include additional channels without affecting the overall infrastructure of our activity recognition system.

4 Evaluation

The experiments were performed by 4 subjects in a virtual reality home environment[1]. 10 activities were included as shown in Table 1. The activities were performed 20 times as part of various scenarios. A scenario comprises of a few related activities performed in some sequence. For example, we used *breakfast scenario*, *lunch*

[1] The subjects were initially taught how to perform activities in a virtual reality environment and then given a time period to practice in this environment. Only when the subjects were comfortable with the environment, they were allowed to perform the activities.

scenario, *free-time scenario*, etc. Some activities were common for several scenarios like the activity of *doing the dishes* which is common to both the *lunch scenario* and the *breakfast scenario*. The subjects were allowed to perform the activities in their own style (often in many different ways).

Table 1. List of actions and activities based on the AMPS framework [20].

	Activity # and name	**Actions within individual activities**
List of 10 activities	1 - Preparing rice	Get the rice bag, Pour rice into the cooker, Pour water into the cooker, Add salt, Put back the rice bag
	2 - Preparing fried vegetables	Get some vegetables, Cut those vegetables, Fry those vegetables, Add spices, Place the chopper in the sink
	3 - Preparing cake	Get the baking plate, Add some eggs, Add some milk, Add some sugar, Add some cake powder, Place the baking plate in the oven
	4 - Preparing coffee	Take some coffee powder, Pour water into the coffee machine, Get some cups, Pour some coffee into the cups
	5 - Preparing breakfast	Toast some bread slices, Boil some eggs, Prepare some juice, Prepare the cereals
	6 - Doing the dishes	Clean the dishes, Dry the dishes on the rack, Wash the hands
	7 - Having lunch	Have the main meal, Have the dessert, Drink coffee, Place the used dishes in the sink
	8 - Having breakfast	Have the main meal, Place the used dishes in the sink
	9 - Preparing the table	Place the table mats, Get some cutlery, Get some plates, Get some glasses, Get the food, Place some napkins
	10 - Cleaning the kitchen	Clean the table, Clean the stove, Clean the rack, Clean the floor

When a subject begins performing his/her activity, each object is in the location where it was last placed in the subject's previous activities. This makes our experiments realistic compared to having a fixed initial location for each objects. Cases when the subjects dropped an object on the floor or grabbed the wrong object were also included in our dataset. A real chair was used for the subjects to perform the activities of *having breakfast* and *having lunch.* Subjects' body postures and locomotion within the VR environment were realistic. For instance, the subjects were not allowed to pass through a table, even though it is possible in a VR environment.

The *number of clusters* and the *observation sequence length* were empirically determined for individual information channels based on the recognition accuracy. We have trained and tested the system on recorded data, using various combinations of these parameters. The optimal parameters are not sharply defined, allowing a variation of around 10% without altering the results significantly. Refer to Table 2.

Table 2. Optimal parameters for "A", "B" and "C".

	Observable Space ("A")	**Manipulable Space ("B")**	**Object Manipulation ("C")**
Number of clusters	55	70	No clusters
Sequence length	15	15	7

4.1 Precision, Recall and Confusion Matrix

We used the "Leave-One-Out Cross-Validation" (LOOCV) scheme to obtain the precision and recall figures (see Table 3). We define precision and recall as follows:

$$precision = \frac{TruePositives}{TruePositives + FalsePositives} \qquad recall = \frac{TruePositives}{TruePositives + FalseNegatives}$$

Table 3. Precision (P) and recall (R) in percentage (%) for each activity (Act) and action (An) using the three information channels (A, B, and C). The last column represents the precision (P) and the recall (R) obtained by combining the three information channels. The last row represents global values (G) in percentage (%).

Act #	Observable space				Manipulable space				Object manipulation				Combination			
	Activity		Action		Activity		Action		Activity		Action		Activity		Action	
	P	R	P	R	P	R	P	R	P	R	P	R	P	R	P	R
1	85	100	70	100	80	100	62	100	100	100	70	100	100	100	83	99
2	80	100	61	100	63	100	40	100	91	99	55	98	93	100	76	99
3	79	99	43	99	69	100	34	99	94	88	73	84	89	98	72	95
4	72	100	60	100	69	100	58	100	73	92	49	86	76	96	67	92
5	80	98	69	98	76	99	59	99	87	94	78	94	92	98	84	98
6	84	94	67	92	79	93	44	88	76	89	51	84	93	99	74	97
7	86	99	49	96	90	100	50	99	65	94	49	93	90	100	67	91
8	81	97	68	95	83	98	77	97	56	94	38	92	81	98	74	98
9	85	98	80	97	83	99	67	98	80	100	73	100	91	96	87	96
10	82	99	76	99	78	97	68	96	65	97	52	95	83	97	76	96
G	**81**	**98**	**64**	**98**	**77**	**99**	**56**	**98**	**79**	**95**	**59**	**92**	**89**	**98**	**76**	**96**

Considering the information channels independently, observable space (OS) shows the best results, both at the activity-level and action-level. A deeper analysis of the clusters formed within OS shows that it contains more than simple situative information. For the same OS (e.g. *"Around the stove"*), several clusters are formed for different activities (e.g. *"Stove + objects for cooking"*, *"Stove + ingredients for preparing cake"*). The clusters actually contain information about both the situation (location) and the activity, considering the objects brought by the user while performing the activity. So the clusters formed in OS are not only snapshots of patterns in the environment, they also represent the dynamic changes done by the user when he/she performs the activities. This also explains why the optimum number of clusters is high for both OS and MS. MS works in a similar manner, though within a smaller space.

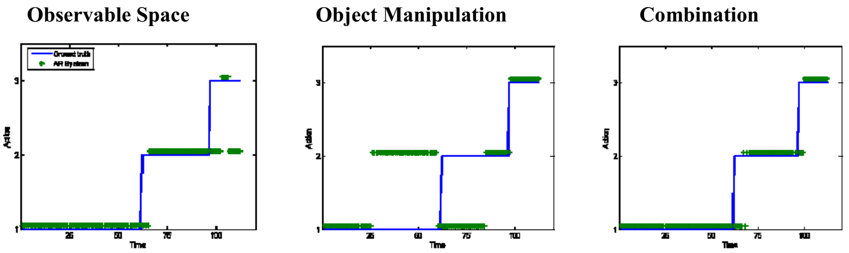
From Table 3 it is also clear that the set of objects in close vicinity to the user (MS) is an important clue to the user's activitiy. Recognition using this information produces results at the activity-level as well as at the action-level similar to object manipulation (OM), which has been a dominant approach in recognizing human activities [10], [11]. For certain activities (1, 2, 3, 5, 9), the actions are recognized better using OM while there is another subset of activities for which the actions are recognized better using MS. This is not just a coincidence since activities 1, 2, 3, 5 and 9 are precisely the "prepare" activities, like for instance *preparing cake* or *preparing the table*, wherein a set of objects has to be picked up from different places and brought to a particular place. On the other hand, manipulable space is more efficient for activities like *having lunch* or *having breakfast*, in which a set of objects is already present at the beginning of the activity and remain present until the end of the activity. The principle of closeness discussed earlier is applicable here where for instance during the action *have the dessert*, the user pushes away the main meal and brings the fruit bowl closer, then pushes the fruit bowl away and brings the coffee jar and cups closer. Refer to Table 4 for confusion matrix.

Table 4. Confusion matrix.

		Recognized Activities									
		1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
Actual Activities	**1**	684	0	0	0	0	0	0	0	0	0
	2	241	3174	0	0	0	0	0	0	0	0
	3	6	119	1187	0	11	0	0	0	0	0
	4	0	18	231	1128	0	0	0	0	0	33
	5	20	6	1	133	2715	0	0	0	0	30
	6	0	0	6	0	171	2430	0	8	0	0
	7	0	0	0	60	0	137	3967	0	298	0
	8	0	0	0	0	0	0	83	2068	344	48
	9	0	0	0	0	4	0	26	166	2529	50
	10	0	0	4	0	142	13	23	1	352	2643

4.2 Information Channels Complement Each Other

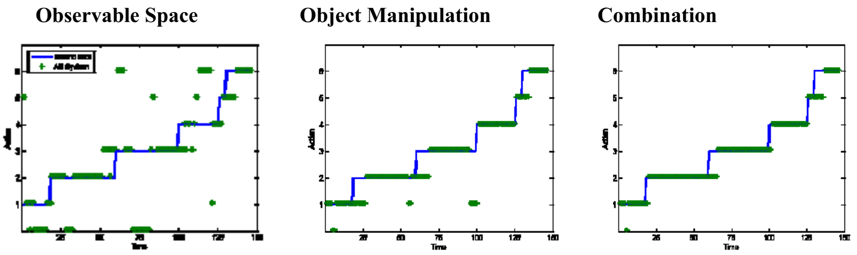
By combining all three information channels, we obtain a recognition precision of 89% at activity level and 76% at action level. Such a high precision with fine granularity at the action level is possible due to the combination of the information channels that represent different and complementary aspects of the user activities. In Fig. 5, for the activity of *doing the dishes*, action 1 (*clean the dishes*) and 2 (*dry the dishes on the rack*) share similar object manipulation data. However, the separation is improved by using observable space, which includes the *sink* for action 1 and the *rack* for action 2.



Activity - Doing the dishes: (1) Clean the dishes - (2) Dry the dishes on the rack - (3) Wash the hands

Fig. 5. Observable space as the dominant information channel for activity recognition.

In some activities, like for instance *preparing fried vegetables*, the actions take place in a fixed location, and the set of objects remains the same because all the objects needed for this activity are already around the user. In such cases, observable space data produces less discriminating information compared to object manipulation at the action-level (see Fig. 6).


Observable Space

Object Manipulation

Combination

Activity - Preparing fried vegetables: (1) Get some vegetables – (2) Cut those vegetables – (3) Fry those vegetables – (4) Add spices – (5) Put the chopper in the sink – (6) Put back the spices (extra action performed sometimes).

Fig. 6. Object manipulation as the dominant information channel for activity recognition.

Activity recognition approaches like [10] and [11] that focus only on object-manipulation patterns encounter difficulties in classifying activities that involve similar sets of objects. For example, [11] keeps track of the number of times an object was touched to differentiate between the activity of *preparing the table* and the activity of *having breakfast* that uses a similar set of objects. Such an approach requires many parameters that need to be fine-tuned for individual cases, thereby limiting its scalability to a variety of activities.

5 Discussion and Future Work

Transferability to Real-World Applications. VR simulations offer “clean” sensing with little noise and uncertainty in the collected signals, as opposed to real-world sensing. OM and MS require the identification of objects close to the actor’s body which can be reliably obtained using RFID technology [21] as shown in many applications including [16], [10] and [11]. MS would require an RFID reader with higher power compared to OM due to the larger reading range necessary. Sensing OS is instead more complicated than OM and MS due to limitations in currently available sensing technologies. However, our VR experiment shows a limited contribution of that space (approximately 4%) to the overall activity recognition precision and we have therefore chosen to ignore OS in our current hardware development activities.

Scalability to Number of Activities. Human actors may require assistance for a potentially large number of activities. Since we use individual models for each activity, our approach should be able cope.. But by increasing the number of activities, the recognition accuracy might decrease. The 10 activities used for evaluating our system all took place in a kitchen environment. We estimate that the total number of activities of the same kind, in the same kind of environment, and at the same level of granularity, is less than 10 times as many. We think our approach will be able to handle such a scaling factor. Furthermore, we believe that although the total repertoire of everyday activities for a person may be large, they may be fairly well distributed over different environments. This would mean a moderate number of possible activities for each environment that need to be distinguished between.

Adaptation to Variations in Activity Patterns. Human actors may perform the same activity in several different ways. We have addressed this issue to some extent by including those variations in the training data. However with time, the actors may change the underlying structure of some activities. Thus, a modelling approach capable of dynamically changing its internal model with time would be desirable, but represents a major challenge. We are currently investigating possible techniques to enhance the capabilities of our HMMs modelling activities to be adaptive with time.

Handling Interrupted and Interleaved Activities. Human activities may be interrupted or interleaved with another activity, which is to some extent addressed in [11]. By introducing states within individual objects to keep track of "key" object state changes, we hope to be able to improve our approach in recognizing the end of an action/activity and thus, the handling of interrupted/interleaved activities.

6 Conclusions

In this paper we have presented a prototype activity recognition system developed on the basis of an "egocentric" view of how an actor interacts with objects to perform everyday activities. When evaluated in a VR-simulated home environment 1) activity and action recognition accuracies using OS and MS has shown promising results comparable to OM-based approaches [10], [11]; 2) fine-grained activity recognition at the action-level has also been demonstrated using the combination of all three information channels (OS, MS, OM). Our work provides an activity-aware platform for further investigations into the development of personal and user-defined activity-assistive systems. We have also demonstrated the utility of a VR-based development methodology when developing ubiquitous and wearable computing systems.

Acknowledgements

We would like to thank Gösta Bucht, Björn Sondell, Kenneth Bodin, Anders Backman and Marcus Maxhall from Umeå University, Sweden. This work is partially funded by the EC Target 1 structural fund program for Northern Norrland, Sweden.

References

1. Starner, T. The Challenges of Wearable Computing: Part 1 & 2, IEEE Micro 21(4), (2001) 44-52 and 54-67
2. B. Nardi. (ed.): Context and Consciousness: Activity Theory and Human-Computer Interaction. Cambridge: MIT Press, (1995)
3. Bao, L., Intille, S. Activity Recognition from User-Annotated Acceleration Data, Second International Conference, PERVASIVE 2004, Linz/Vienna, Austria. LNCS 3001, April (2004) 1-17
4. S.W. Lee and K. Mase. Activity and location recognition using wearable sensors. IEEE Pervasive Computing 1(3), (2002) 24-32

5. N. Kern, B. Schiele, A. Schmidt. Multi-Sensor Activity Context Detection for Wearable Computing, European Symposium on Ambient Intelligence, (2003)
6. Chen, J., Kam, A., Zhang, J., Liu, N., Shue, L. Bathroom Activity Monitoring Based on Sound, Third International Conference, PERVASIVE 2005. LNCS 3468, April (2005) 47-61
7. Haritaoglu, I., Harwood, D., Davis, L. W^4: Real-Time Surveillance of People and Their Activities, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 22, No. 8, August (2000)
8. Lukowicz, P., Ward, J., Junker, H., Stäger, M., Tröster, G., Atrash, A, Starner, T. Recognizing Workshop Activity Using Body Worn Microphones and Accelerometers, Second International Conference, PERVASIVE 2004, Linz/Vienna, Austria. LNCS 3001, April (2004) 18-32
9. J. Lester, T. Choudhury, N. Kern, G. Borriello, B. Hannaford. A Hybrid Discriminative/Generative Approach for Modeling Human Activities. 19th International Joint Conference on Artificial Intelligence, (2005)
10. Philipose, M., Fishkin, K., Perkowitz, M., Patterson, D., Fox, D., Kautz, H., Hähnel, D. Inferring Activities from Interactions with Objects, IEEE Pervasive Computing, October (2004) 50-57
11. Patterson, D., Fox, D., Kautz, H., Philipose, M. Fine-Grained Activity Recognition by Aggregating Abstract Object Usage, Ninth IEEE International Symposium on Wearable Computers, (2005)
12. Suchman, L. Plans and situated actions: the problem of human machine interaction. Cambridge: Cambridge University Press, (1987)
13. Janlert, L.-E. Putting Pictures in Context. Proceedings of AVI2006, ACM Press, (2006) 463-466
14. Tapia, E., Intille, S., Larson, K. Activity Recognition in the Home Using Simple and Ubiquitous Sensors, Second International Conference, PERVASIVE 2004, Linz/Vienna, Austria. LNCS 3001, April (2004) 158-175
15. Pederson, T. From Conceptual Links to Causal Relations — Physical-Virtual Artefacts in Mixed-Reality Space. PhD thesis, Dept. of Computing Science, Umeå university, report UMINF-03.14, ISBN 91-7305-556-5, (2003)
16. Pederson, T. Magic Touch: A Simple Object Location Tracking System Enabling the Development of Physical-Virtual Artefacts in Office Environments. Journal of Personal and Ubiquitous Computing, Vol. 5. Springer (2001) 54-57
17. Backman, A. Colosseum3D – Authoring Framework for Virtual Environments. In Proceedings of EUROGRAPHICS Workshop IPT & EGVE Workshop, (2005) 225-226
18. Fritzke, B. A growing neural gas network learns topologies. In G.Tesauro, D.S. Touretzky, and T.K. Leen. (ed.): Advances in Neural Information Processing Systems 7, MIT Press, Cambridge MA, (1995) 625-632
19. Rabiner, L. A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition. Proceedings of the IEEE, Vol. 77, No. 2, February (1989)
20. AMPS. http://www.ampsintl.com/ as on 2nd January (2007)
21. Finkenzeller, K. *RFID Handbook.* John Wiley and Sons, New York, NY, USA, Second edition, (2003)