Behaviour Recognition from Sensory Streams in Smart Environments

Sook-Ling Chua, Stephen Marsland, and Hans W. Guesgen

School of Engineering and Advanced Technology, Massey University,
Palmerston North 4442, New Zealand
{s.l.chua,s.r.marsland,h.w.guesgen}@massey.ac.nz
http://muse.massey.ac.nz/

Abstract. One application of smart homes is to take sensor activations from a variety of sensors around the house and use them to recognise the particular behaviours of the inhabitants. This can be useful for monitoring of the elderly or cognitively impaired, amongst other applications. Since the behaviours themselves are not directly observed, only the observations by sensors, it is common to build a probabilistic model of how behaviours arise from these observations, for example in the form of a Hidden Markov Model (HMM). In this paper we present a method of selecting which of a set of trained HMMs best matches the current observations, together with experiments showing that it can reliably detect and segment the sensor stream into behaviours. We demonstrate our algorithm on real sensor data obtained from the MIT PlaceLab. The results show a significant improvement in the recognition accuracy over other approaches.

Keywords: Behaviour Recognition, Hidden Markov Models (HMMs), Activity Segmentation, Smart Home.

1 Introduction

It is a well-reported fact that the populations of the Western world are aging. In Europe, for example, the number of people aged 65 and over is projected to increase from 10% of the entire population in 1950 to more than 25% in 2050. Older adults are more frequently subject to physical disabilities and cognitive impairments than younger people. It is clearly impossible to rely solely on increasing the number of caregivers, since even now it is difficult and expensive to find care. Additionally, many people are choosing to stay in their own homes as long as possible, and hope to remain independent. This has lead to a large number of monitoring systems (also known as 'smart homes', or 'ubiquitous computing systems') that aim to assist in the Activities of Daily Living (ADLs) such as bathing, grooming, dressing, eating and so on [11], either directly through involvement with the person, or by alerting carers when a problem arises.

As the majority of the ADLs involve using physical objects, such as washing machines, cooking utensils, refrigerators, televisions and so forth, it is possible to infer the inhabitant's behaviour [9], [13]. As a result, behaviour recognition has been drawing significant attention from the research community. The idea behind behaviour recognition is to infer the inhabitant's behaviours from a series of observations acquired through sensors.

A. Nicholson and X. Li (Eds.): AI 2009, LNAI 5866, pp. 666–675, 2009.
© Springer-Verlag Berlin Heidelberg 2009

One of the main challenges in behaviour recognition is that the exact activities are not directly observed. The only information provided are the sensor observations, which could be that the kitchen light is on, the oven is turned on and the burner is on; the inference that therefore somebody is cooking is left to the intelligent part of the system. Two challenges of behaviour recognition are that many of the same sensor activations will be involved in multiple behaviours, and that the number of observations in a behaviour can vary between activities, and within different instances of the same activity. For example, making breakfast could involve sensors on the fridge, toaster, and cabinet one day, and also the kettle the next day when the person decides to have coffee as well. Making lunch will also involve the fridge and cabinet, and other unrelated sensors.

One common approach to recognising behaviours is to use Hidden Markov Models (HMMs), which are probabilistic graphical models where sensor observations give rise to latent variables which represent the behaviours. To use HMMs there are a few problems that have to be solved. One is to break the token sequence into appropriate pieces that represent individual behaviours (i.e., segmentation), and another is to classify the behaviours using the HMM. Most current approaches assume that the activities have been segmented, and use a fixed window length to partition the input stream. With each behaviour produces different numbers of sensor actions, it is inappropriate to rely on fixed window length, as activity segmentation can be biased in this way. Thus, an intelligent method is required to self-determine the window size based on the data. This paper presents a prototype system that performs the behaviour recognition and segmentation by using a set of HMMs that each recognise different behaviours and that compete to explain the current observations. In this paper we propose a variable window length that moves over the sequence of observations and use hand-labelled data to demonstrate the efficacy of the system.

2 Related Work

There has been a lot of work on activity segmentation in smart homes. Within smart home research it is common to use more complicated variants of the HMM, such as the Hierarchical Hidden Markov Model [7], or Switching Hidden Semi-Markov Model [1]. In both of these models, a top-level representation of behaviours (e.g., cooking or making coffee) is built up from a set of recognised activities that arise from the individual sensor values. A variant of these methods uses a three level Dynamic Bayesian Network [5] (the HMM is one of the simplest dynamic Bayesian network). These models can be seen as adding complexity to the HMM in order to represent one complete model of behaviours arising from sensor activations. The difficulty with these methods is that more complex models require more data for training, and higher computational cost.

There are many other places where time series of activities are recognised and classified into ‘behaviours’, and our method owes more to other areas of temporal signal analysis, such as recognising activities from posture information from video [3] and motion patterns [8,4,14]. In common with our algorithm, Kellokumpu, Pietikäinen and Heikkilä [3] use a set of HMMs, one for each activity, and apply the forward algorithm in the same way that we do to monitor likelihood values. However, they do not use a sliding time window, preferring multiple window sizes and thresholding in order to separate out the activities. Niu and Abdel-Mottaleb [8] merge the outputs of the different

HMMs using majority voting. A vote is assigned to each window and activity is classified based on the most common classification from the set of HMMs. A similar method is used in [12] for the identification of housekeeping activities using RFID data.

Kim, Song and Kim [4] turn the problem around and perform segmentation before classification, in this case for gesture recognition. The starting point of gestures is detected, and then a window is slid across the observation sequence until an end point is reached. The extracted gestures are then fed to HMMs for gesture recognition, with the final gesture type being determined by majority vote. An attempt to simultaneously detect the sequences and train the HMMs was described by Yin, Shen, Yang and Li [14]. A window is moved over the observation sequence to construct a linear dynamic system, and the likelihood of each model is computed based on these linear systems. A modified EM algorithm is used to simultaneously update these estimates. High-level goals can then be inferred from these sequences of consecutive motion patterns. Another method that is closely related to our approach is the work of Govindaraju and Veloso [2], which attempts to recognise activities from a stream of video. They use a set of HMMs, but maintain a single fixed window size, which is determined by averaging the lengths of the training segments used.

3 Behaviour Recognition

In our work we use the Hidden Markov Model as the basic representation of a behaviour. We posit that a typical behaviour is a sequence of activities that occur close to one another in time, in one location. While this is not always true, for now we are focussing on these types of behaviour, which includes activities such as cooking and preparing beverages. It would not include common activities such as laundry, which may well be separated in time (while waiting for the washer to finish) and in space (for example, if clothes are hung outside rather than using a dryer).

The Hidden Markov Model (HMM) [10] is very commonly used for these types of problem. It is a probabilistic model that uses a set of hidden (unknown) states to classify a sequence of observations over time. The HMM uses three sets of probabilities, which form the model parameters: (1) state transition probability distribution $A = a_{ij}$, the probability of transition from state i to state j conditional on current state i, (2) observation probability distribution $B = b_j(O)$, which illustrates the probability of observing observation O given that current state is j and (3) initial state distribution $\pi = \pi_i$. The HMM is a special case of the Dynamical Bayesian Network or Graphical Model [6], and unlike most graphical models, HMMs admit tractable algorithms for learning and prediction without the need for sampling or approximation. We use a separate HMM to recognise each behaviour. This is allows for variation in the activity, such as different orders of sensor activation, the fact that certain sensor activations can be shared by multiple behaviours, and the fact that the algorithm is probabilistic and can hence deal with 'noise' in the data.

Given a set of HMMs trained on different behaviours, we present data from the sensor stream to all of the HMMs, and use the forward algorithm [10] to compute the likelihood of this sequence of activities according to the model of each behaviour, i.e. $P(O_1, O_2, \ldots, O_T|\lambda)$, for HMM λ and observation sequence $O_1, O_2, \ldots, O_T$ using:

$$P(O_1, O_2, \ldots, O_T|\lambda) = \sum_{i=1}^{N} \alpha_T(i)$$

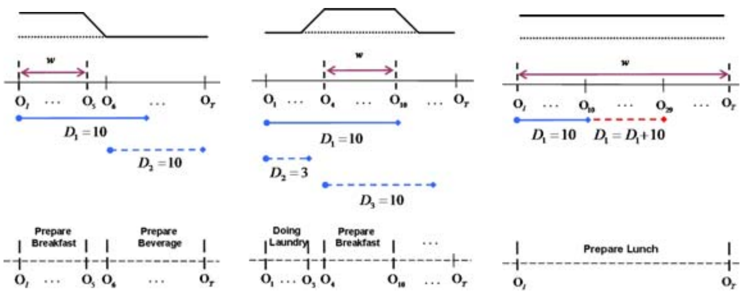
which can be recursively computed by:

$$\alpha_1(i) = \pi_i b_i(O_1)$$
$$\alpha_{t+1}(j) = \sum_{i=1}^{N} \alpha_t(i) a_{ij} b_j(O_{t+1})$$

The α_t values are quantised into the set $\{0, 1\}$ using a winner-takes-all approach to simplify the calculations at each subsequent step. The forward algorithm fits well into the context of our study because it determines how well the 'winning' HMM explains the observed sequences. This can be determined by monitoring the forward variable (α) for each observation. A change in the quantised α value signifies a 'change' of activity from the observation stream. The data that is presented to the HMMs is chosen from the sensor stream using a window that moves over the sequence. The choice of the size of this window is important, because it is unlikely that all of the activities in the sequence belong to one behaviour, and so the HMM chosen to represent it will, at best, represent only some of the activities in the sequence. Many of the methods described in related work used multiple sizes of window to try and deal with this fact, which arises because sequences of different behaviours (or indeed, the same behaviour in different instances) can be of different lengths. We present an alternative solution to this problem. To see the importance of the problem, consider the three different cases shown in Fig. 1. In each, a behaviour w takes up much of the window and is the winning behaviour. However, the location of it in the window differs, and we want to ensure that other behaviours in the window are also recognised.

The solid line shown in Fig. 1 illustrates how the quantised α values computed by the forward algorithm applied to one particular HMM, the one selected as the 'winner' for this window. If the quantised α values are high (that is, $\alpha = 1$) at the beginning of the observation sequence then it is likely that case (Fig. 1(a)) is occurring. Following Fig. 1(a) we see that there is a drop in α value between observations O_5 and O_6, which suggests that the behaviour has changed. We can therefore classify $O_1, O_2, \ldots, O_5$ as belonging to the winning behaviour, w, and then initialise a new window of default size (D_2) at O_6. When D_2 is initialized, all the observations within D_2 will then be fed to HMMs for competition and the process iterates. The second case occurs when the winning behaviour best describes observations that fall in the middle of the window, e.g., $O_4, O_5, \ldots, O_{10}$ in Fig. 1(b). Since the winning behaviour (w) does not describe observations O_1, O_2 and O_3, the probability for these three observations is low and we observe a jump in the α value at O_4. When this is observed, a new window (D_2) is initialized that contains only the three observations that are not explained by behaviour w. The whole process is then recursively computed on this window. With regard to the remaining sequence (O_4 and onwards) it would be possible to use HMM w and continue to monitor the α values. However, it was observed that sometimes there may be an overlap in individual sensor activations between the first and second behaviours, which can confuse things. For this reason, a new window of default size (D_3) is started at O_4 and the HMM competition is rerun on this sequence.

Since these two cases have ensured that the winning behaviour is at the beginning of the window, the only possibilities are that the behaviour stops during the window (Fig. 1(a)) or does not (Fig. 1(c)). The first case is already dealt with, and in the second case, we could simply classify the activities in the window as w and start a new one at the end of the current window. However, instead we extend the size of the window (shown as a dashed arrow in Fig. 1(c)) and continue to calculate the α value for each observation until the α drops. Fig. 2 summarises the overall procedures of the proposed method.



(a) (b) (c)

Fig. 1. An activity w does not need to take up the entire window. Even assuming that the actions in a behaviour are contiguous, it could be (a) at the start of the window, (b) in the middle, or (c) at the end. If the entire window is classified as one behaviour, then a potentially large number of behaviours are missed. $O_1, O_2, \ldots, O_T$ is the observation sequence, D is the window size and the initial default window size is 10. The solid line above the observation sequence shows the possible representations of a winning sequence using the α values. The long dash below the observation sequence shows the original observation sequence. For details, see the text.

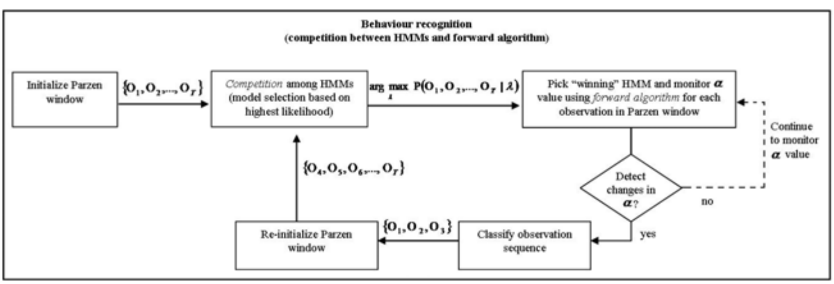


Fig. 2. Summary of our algorithm. When no changes is observed in α value, the algorithm will continue to monitor the α value based on the winning HMM (shown in dashed line). The recognition process is recursively computed until it reaches the end of the observation stream.

4 Experiment and Results

In order to demonstrate our algorithm, we took a dataset from the MIT PlaceLab [13]. They designed a system based on a set of simple and easily installed state-change sensors that were placed in two different apartments with real people living in them. The subjects kept a record of their activities that form a set of annotations for the data, meaning that there is a ‘ground-truth’ segmentation of the dataset. We trained the HMMs using this hand-segmented and labelled data. While this is a simplification of the overall aims of the project, it enables us to evaluate the method properly; future work will consider the problems of training with noisy and unlabelled data.

The actual dataset consists of state changes in objects within the home (such as the washing machine, TV, coffee machine, and toaster). For the first of the two subjects there were 77 sensors and data was collected for 16 consecutive days. It is this dataset that will form the basis for most of the experiment reported here. Further details on these datasets and PlaceLab architecture can be found in [13]. We assume for now that activities take place in one room, and that the location of the sensors is known *a priori*. For this reason, we concentrated on just one room, namely the kitchen, which contained more behaviours than any other. The behaviours that were originally labelled in the kitchen were (i) prepare breakfast, (ii) prepare beverage, (iii) prepare lunch, and (iv) do the laundry. We split behaviour (i) into two different ones, prepare toast and prepare cereal. This made two relatively similar behaviours, which is important to test recognition accuracy to distinguish activities and to avoid bias classification.

In order to train the HMMs, a subset of the data was required. We partitioned the data into a training set consisting of the first few days, followed by a test set consisting of the remainder. From the total of 16 days of data, we tried different splits of the data, from 15 days for training (and 1 for testing) through 11 days, 8 days, and 5 days for training. There were approximately 5-6 activities each day, made up of around 90-100 sensor observations. The HMMs were each trained on the relevant labelled data in the training set using the standard Expectation-Maximization (EM) algorithm [10]. We conducted three separate experiments using these five trained HMMs. In the first, we compared the algorithm with fixed window length, while in the second we looked at the effects of window size on the efficiency and accuracy of the algorithm. In the third experiment, we looked at how much training data was required for accurate results.

We defined two separate measurements of accuracy for our results:

Behaviour-level recognition accuracy: This simply compares the behaviour output by the algorithm with that of the label whenever the behaviour changed (e.g. behaviour such as ‘doing laundry’, ‘preparing lunch’, etc.).

Observation-level recognition accuracy: This compares the behaviour output by the algorithm with that of the label for every observation. This is particularly sensitive to times when two behaviours that occur one after the other share the same observations (e.g. observation such as ‘oven is turned on’ should be classified as ‘preparing lunch’ rather than ‘doing laundry’).

4.1 Experiment 1: Comparison between the Algorithm with Fixed Window Length

The first experiment is designed to compare the algorithm with the fixed window length. In this experiment, we used a fixed window length of size 10, with 5 days of training and 11 days of testing, and ran the algorithm over the sensory stream. Table 1 shows

Table 1. Comparison results between the variable window length and fixed window length

Recognition Accuracy	Variable Window Length	Fixed Window Length
Behaviour-level	87.80%	78%
Observation-level	98.35%	86.93%

the comparison results between the algorithm with the fixed window length. Results are based on contiguous manner, so there is no variation when the experiment is rerun, hence averages and standard deviations are not reported.

4.2 Experiment 2: Competition among HMMs

Before beginning this experiment, we used a window size of 10, with 5 days of training and 11 days of testing, and ran the entire algorithm over the sensor observations. The results of sliding this window over the data is shown in Fig. 3, which displays the outputs of the algorithm, with the winning behaviour at each time being clearly visible. As the figure shows, we can determine that the subject is doing laundry at observation

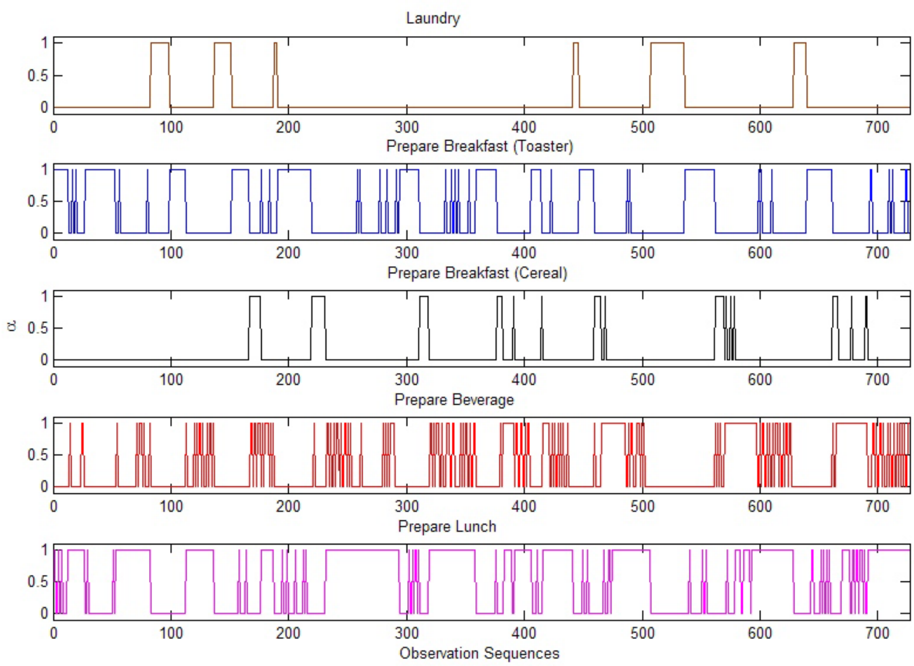


Fig. 3. Illustration of competition between HMMs based on a testing set of 727 sensor observations for five different behaviours: laundry, preparing toaster, preparing cereal, preparing beverage and preparing lunch. Since behaviours may share the same sensor observation, this explains why the $\alpha = 1$ is seen in multiple behaviours (e.g. between observation 120 and 140 in the last two behaviours). The 'preparing lunch' is selected as winner because it appears in a continuous manner.

150 and preparing breakfast (toaster) at observation 550. The classification accuracy of this experiment was high enough to encourage us to look further.

Table 2 shows the results of using different lengths of window. The different in the results is not significantly different across the different sizes, and therefore a shorter window length is preferred in order to keep the computational costs low.

Table 2. The results of using different initial window length on different training–test sets

Training Set (days)	**Test Set** (days)	**Initial Window Length**	**% Accuracy (Behaviour)**	**% Accuracy (Observation)**
11	5	10	86.96	97.29
		20	86.96	97.29
		50	86.96	97.29
		100	86.96	97.29
8	8	10	88.23	98.14
		20	82.35	97.80
		50	82.35	97.80
		100	88.23	98.14
5	11	10	87.80	98.35
		20	82.93	98.07
		50	82.93	98.07
		100	82.93	98.35

4.3 Experiment 3: Size of Training Data

The objective of this experiment is to analyze the amount of training data needed to train the HMMs. The most important thing is that every behaviour is seen several times in the training set to ensure that the HMM acquires a good representation of that behaviour. The results on recognition accuracy on both behaviour-level and observation-level are presented in Table 3. As the table shows, the size of training data does not have much significant impact on recognition accuracy. Even when only 5 days of training and 11 days of testing with window size 10 are used, we are still able to achieve 87.80% recognition accuracy on behaviour-level and 98.35% on observation-level. It seems that the proposed method does not need a significant large amount of training data for this dataset, although this may not be true for more complicated behaviours.

Table 3. Behaviour-level and observation-level recognition accuracy using window length of size 10

		Behaviour-level		**Observation-level**	
Training Datasets	**Test Datasets**	**Total Activities**	**Accuracy**	**Total Observations**	**Accuracy**
15 Days	1 Day	5	100%	99	100%
11 Days	5 Days	23	86.96%	369	97.29%
8 Days	8 Days	34	88.23%	591	98.14%
5 Days	11 Days	41	87.80%	727	98.35%

5 Discussion

On this relatively simple dataset our algorithms have worked very well, producing over 98% accuracy at the observation-level. However, it is still instructive to see if there are consistent reasons for the misclassifications that did occur.

We identified one main reason for misclassification, which is that individual sensor observations can be in several behaviours. There are two places where this can be a problem. The first is when the end of one behaviour contains observations that could be in the start of the next. This will not pose a problem if the second behaviour happens immediately after the first. However, if the second behaviour happened two hours after the first, that would be a totally different unrelated behaviours. The second place that this can be seen is where the winning behaviour is not at the start of the window, but those activities at the start could be interpreted as being part of that behaviour. It was experimentally observed that this was more likely to happen where the size of the window was large, because more behaviours were observed.

One way to reduce the misclassification is by adding extra information in order to improve the classification accuracy. This can be achieved by augmenting the current algorithm with spatio-temporal information. If spatio-temporal information is included, then places where two behaviours abut one another can be reduced, since there could be other non-kitchen behaviours inbetween.

6 Conclusions

We have presented a simple system that performs behaviour recognition based on competition between trained Hidden Markov Models, and demonstrated that the method works on labelled data. Our experimental results show that the method works effectively, with an average of around 90.75% behaviour-level recognition accuracy and 98.45% observation-level recognition accuracy (by averaging the accuracy percentage from table 3) based on relatively small amount of training data. We have investigated the size of window required, and found that relatively small ones work best, which reduces the amount of training data required even further. As the model is relatively simple and based on recursive computation, the computational costs are significantly lower than many other methods. We have also shown that a comparison between variable window length and fixed window length and that the variable window length works best.

It is important to note that this study is purely performed on labelled data and have proven the ability to distinguish activities given a series of observations. The encouraging results highlight the need to test on unlabelled data, resulting in a system that can be built up from nothing when sensors are placed into a new environment, and allowing on-line recognition. The MIT PlaceLab dataset is very clean, in that there is little sensor noise or inaccuracy. This may well not be the case with other datasets, since sensors can be 'twitchy' or fail, there may be other people or animals in the house, etc. It is possible that smoothing the sensor stream will deal with this, e.g., by using a median filter. It may also be that behaviours are interleaved: a person may well make a beverage at the same time preparing lunch, which could be done while the laundry was running. Our current system will not deal with these behaviours in any sensible way, highlighting all of the separate parts of the behaviour as different instances of that behaviour. This is left for future work.

Acknowledgments. This study is conducted as part of PhD research in the School of Engineering and Advanced Technology at Massey University. The financial support from Massey University is gratefully acknowledged. We also acknowledge the support of the other members of the MUSE group (`http://muse.massey.ac.nz`). We would like to thank MIT PlaceLab for providing access to their dataset.

References

1. Duong, T.V., Bui, H.H., Phung, D.Q., Venkatesh, S.: Activity recognition and abnormality detection with the switching hidden semi-markov model. In: CVPR 2005: Proc. of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2005), vol. 1, pp. 838–845. IEEE Computer Society, Los Alamitos (2005)
2. Govindaraju, D., Veloso, M.: Learning and recognizing activities in streams of video. In: Proc. of the AAAI Workshop on Learning in Computer Vision (2005)
3. Kellokumpu, V., Pietikäinen, M., Heikkilä, J.: Human activity recognition using sequences of postures. In: MVA, pp. 570–573 (2005)
4. Kim, D., Song, J., Kim, D.: Simultaneous gesture segmentation and recognition based on forward spotting accumulative hmms. Pattern Recognition 40(11), 3012–3026 (2007)
5. Liao, L., Patterson, D.J., Fox, D., Kautz, H.: Learning and inferring transportation routines. Artificial Intelligence 171(5-6), 311–331 (2007)
6. Marsland, S.: Machine Learning: An Algorithmic Introduction. CRC Press, New Jersey (2009)
7. Nguyen, N., Phung, D., Venkatesh, S., Bui, H.: Learning and detecting activities from movement trajectories using the hierarchical hidden markov model. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2005), vol. 2, pp. 955–960 (2005)
8. Niu, F., Abdel-Mottaleb, M.: HMM-based segmentation and recognition of human activities from video sequences. In: IEEE International Conference on Multimedia and Expo. (ICME 2005), pp. 804–807 (2005)
9. Philipose, M., Fishkin, K., Perkowitz, M., Patterson, D., Fox, D., Kautz, H., Hahnel, D.: Inferring activities from interactions with objects. IEEE Pervasive Computing 3(4), 50–57 (2004)
10. Rabiner, L.: A tutorial on hidden Markov models and selected applications in speech recognition. Proc. of the IEEE 77(2), 257–286 (1989)
11. Robert, N., Taewoon, K., Mitchell, l., Stephen, K.: Living quarters and unmet need for personal care assistance among adults with disabilities. Journal of Gerontology: Social Sciences 60B(4), S205–S213 (2005)
12. Stikic, M., Huỳnh, T., Van Laerhoven, K., Schiele, B.: ADL recognition based on the combination of RFID and accelerometer sensing. In: Second International Conference on Pervasive Computing Technologies for Healthcare, PervasiveHealth 2008, pp. 258–263 (2008)
13. Tapia, E.M., Intille, S.S., Larson, K.: Activity recognition in the home using simple and ubiquitous sensors. In: Ferscha, A., Mattern, F. (eds.) PERVASIVE 2004. LNCS, vol. 3001, pp. 158–175. Springer, Heidelberg (2004)
14. Yin, J., Shen, D., Yang, Q., Li, Z.-N.: Activity recognition through goal-based segmentation. In: Proc. of the 19th AAAI Conference on Artificial Intelligence (AAAI 2005), pp. 28–33 (2005)