An event-based approach to multi-modal activity modeling and recognition

Marten Pijl
User Experiences Department
Philips Research
Eindhoven, The Netherlands
marten.pijl@philips.com

Steven van de Par
Digital Signal Processing Department
Philips Research
Eindhoven, The Netherlands
steven.van.de.par@philips.com

Caifeng Shan
Video Processing & Analysis Department
Philips Research
Eindhoven, The Netherlands
caifeng.shan@philips.com

***Abstract*—The topic of human activity modeling and recognition still provides many challenges, despite receiving considerable attention. These challenges include the large number of sensors often required for accurate activity recognition, and the need for user-specific training samples. In this paper, an approach is presented for recognition of activities of daily living (ADL) using only a single camera and microphone as sensors. Scene analysis techniques are used to classify audio and video events, which are used to model a set of activities using hidden Markov models. Data was obtained through recordings of 8 participants. The events generated by scene analysis algorithms are compared to events obtained through manual annotation. In addition, several model parameter estimation techniques are compared. In a number of experiments, it is shown that if activities are fully observed these models yield a class accuracy of 97% on annotated data, and 94% on scene analysis data. Using a sliding window approach to classify activities in progress yields a class accuracy of 79% on annotated data, and 73% on scene analysis data. It is also shown that a multi-modal approach yields superior results compared to either individual modality on scene analysis data. Finally, it can be concluded the created models perform well even across participants.**

***Keywords*-activity recognition; hidden Markov models.**

I. Introduction

Human activity modeling and recognition has long been a popular topic in the field of pervasive computing, with numerous diverse application domains including well-being, security and social applications. However, despite the amount of attention this topic has received, numerous challenges still remain. First, most solutions proposed require an extensive amount of sensors to be installed, for example sensors in household items or arrays of video cameras. Alternatively, users are required to wear special clothing with sensors embedded. Second, a separate set of training samples must often be acquired for each user. As a result, effective activity recognition of a new user is only possible after a training period, which new users might experience as burdensome.

In this paper, a solution is presented with the aim of addressing these challenges. In particular, the focus is on activities of daily living (ADL) such as eating, cleaning, and so on. ADLs are often complex activities made up of a number of subtasks, and typically take a number of minutes to complete. To address the first challenge, activity recognition is based on a single unit, containing all required sensors. This setup allows for easy installation into an existing room, with no need for extensive wiring or requirements placed on the users. For sensors, a single standard-resolution camera and off-the-shelf microphone were used.

A further aim of this work is to construct activity models which are general enough to be valid across users, eliminating the need to reconstruct models when activity recognition is applied to new users. Hidden Markov models (HMM) and coupled hidden Markov models (c-HMM) are used to model a set of selected activities. As observable input for these models, a number of audio and video 'events' are defined. Events are in-the-moment observations made based on information from a single modality, their nature depending on the sensor and algorithms used. For audio, events represent classified sounds, while for video events represent the location of a user in the camera view. A variety of scene analysis algorithms were used to classify events from raw sensor data. Also, several methods for constructing activity models are examined. An overview of this approach is provided in Figure 1.

To evaluate this approach, 8 participants were asked to perform a set of activities, which resulted in over 4 hours of activities recordings in total. The recordings were then fully annotated with ground truth activities and events. Throughout the paper, accuracy is tested on the annotated data as an indication of the classifying power of the activity models under ideal conditions, as well as on the scene analysis data, which contains more errors as would be expected in a realistic environment. Performance is evaluated in three experiments: first, classification of individual recorded activities. Second, classification on ongoing activities in a recording session using a sliding window of observed events. Third, an examination of multiple versus single modality classifiers.

The remainder of this paper is organized as follows: in Section II related work is discussed. In Section III an overview is given of the methods and algorithms used in this work. The experimental setup used to gather data and perform experiments is discussed in Section IV. The experimental results are detailed in Section V. The results,

978-1-4244-5328-3/09/$25.00 ©2009 IEEE

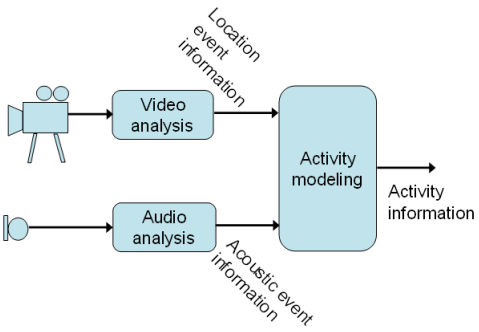

Figure 1. Overview of the approach used in this work. Video and audio events are generated through scene analysis techniques, which in turn provide input for activity modeling.

limitations of the approach and future steps are discussed in Section VI, and a conclusion is provided in Section VII.

II. Related work

The detection and classification of human activities has received considerable attention in the literature. Many studies in this field make use of sensor data obtained from sensors embedded in household objects ([1]), such as RFID tags or pressure sensors, often combined with environmental sensors such as light, motion or temperature ([2]). The disadvantage of such environmental sensor networks is that installation into existing homes is often impractical due to demands on power, wiring, space, or simply due to the amount of sensors to be installed. The use of RFID tags avoids many of these problems, but requires users to carry an RFID tag reader with them. In [3], data from sensors connected to wireless network nodes are used to recognize a number of activities, alleviating the need for extensive wiring.

As an alternative to environmental sensors, wearable sensors are sometimes employed, generally accelerometers attached to various positions on the body ([4]). Accelerometers have also been used in gesture recognition to track hand or body movements ([5]), a task similar to activity recognition. As with RFID tags, users are required to wear one or more pieces of equipment to enable activity recognition.

The use of video images for activity recognition is also well-documented in the literature. An overview of common techniques can be found in [6]. Activity recognition using video images often focuses on posture recognition, such as in [7]. In [8], a dual-camera system for activity recognition is described. There appears to be little work done on activity recognition using audio as a single modality.

Hidden Markov models and a number of extensions on hidden Markov models have been used previously in activity recognition. Hidden Markov models are combined with audio and video to recognize simple office activities such as phone calls and paperwork in [9]. In [10] a hierarchical hidden Markov model combined with low level audio and video features is used for detecting 'action' sequences in movies. Layered hidden Markov models have been used for activity recognition in [11] to detect various office activities. Coupled hidden Markov models were used in [15] to recognize certain T'ai Chi hand gestures. Here, the authors used a vision-based stereo tracking system and model each hand as a seperate process. In the work described here, coupled hidden Markov models are used to model the individual modalities instead.

III. Methods

The hidden Markov model (HMM) has long been a popular tool in modeling temporal processes. A hidden Markov model consists of a number of hidden states, or states for short, with transition probabilities defined between states. A hidden Markov model resides in one of its hidden states, and at given time intervals traverses from its current state to a next state, possibly the same as the current state. Whenever the HMM traverses to a next state, it produces an emission. Unlike the model's state, emissions produced by the HMM can be observed. Hidden Markov models are explained in more detail in [12].

When transitioning between states, the set of transition probabilities together with the current state determine the next state of the model. For each state in the model, there exist a number of transition probabilities from that state to other states (possibly including back to the same state). If q_t is the state of the model at a given time t, and the set of all N hidden states is denoted by $S = \{S_1, S_2, \cdots, S_N\}$, then the transition probabilities $A = \{a_{ij}\}$ can be defined as

$$a_{ij} = P(q_{t+1} = S_j | q_t = S_i), \tag{1}$$

where probability a_{ij} must have the properties that $a_{ij} \geq 0$ for each pair i,j and $\sum_{j=1}^{N} a_{ij} = 1$ for all i. Transition probabilities remain constant over time, that is, they are not affected by the parameter t. As the definition suggests, the next state is dependent solely on the current state. This is called the Markov property.

Similarly, when a state transition is completed, an emission is produced based on a set of emission probabilities. In a model with an alphabet of M emissions $V = \{V_1, V_2, \cdots, V_M\}$, the emission probabilities $B = \{b_j(k)\}$ are defined as

$$b_j(k) = P(V_k | q_t = S_j), \tag{2}$$

where $b_j(k)$ has the properties that $b_j(k) \geq 0$ for each j,k and $\sum_{k=1}^{M} b_j(k) = 1$ for all j. As can be seen from the definition, the observed emission depends on the current state of the model. Therefore, by observing a sequence of emissions, inference can be performed on the model's hidden states.

In activity classification, each activity is represented as a separate HMM. Here, the hidden states represent the process of performing the activity, something not directly

observable by sensors. However, what can be observed are audio and video events detected by the microphone and camera. Accordingly, these observations are modeled as the HMM's emissions. As different activities typically produce a different set of movements and sounds, recognizing activities can be achieved by finding the HMM which best matches the audio and video events observed.

A. The Baum-Welch Algorithm

Unfortunately, given a set of emission sequences, finding the globally optimal values for the model's parameters A and B in acceptable computation time is not possible with current techniques. However, it is possible to optimize the model parameters locally by using e.g. gradient descent methods or the methods described here. As the model parameters can only be optimized locally, it is important to have an appropriate initial model before optimization.

Arguably the best known approach to estimate model parameters is the Baum-Welch (BW) algorithm (e.g. see [12]). The Baum-Welch algorithm is essentially an expectation-maximization algorithm, and works by re-estimating hidden state and emission probabilities A and B over a number of iterative steps. The value $\varepsilon_t(i,j)$ is defined as the probability of transitioning from state S_i at time t, to state S_j at time $t{+}1$, given an observation sequence O of length L. Formally,

$$\varepsilon_t(i,j) = P(q_t = S_i, q_{t+1} = S_j | O) \tag{3}$$

In addition, $\gamma_t(i)$ is defined as the probability of the model residing in state S_i at time t given O. $\gamma_t(i)$ can therefore be written as

$$\gamma_t(i) = \sum_{j=1}^{N} \varepsilon_t(i,j) \tag{4}$$

By summing $\gamma_t(i)$ over all t, the expected number of times the model transitions from state S_i is determined. Similarly, summing $\varepsilon_t(i,j)$ over all t yields the expected number of times the model transitions from state S_i to state S_j. As discussed above, the transition probabilities can be re-estimated by dividing the expected number of transitions from one state S_i to another state S_j by the total expected transitions from state S_i. Therefore, the re-estimation formula for the re-estimated transition probabilities $\bar{a}_{ij}$ is given by

$$\bar{a}_{ij} = \frac{\sum_{t=1}^{L-1} \varepsilon_t(i,j)}{\sum_{t=1}^{L-1} \gamma_t(i)} \tag{5}$$

Similarly, as described above, the re-estimated emission probabilities $\bar{b}_j(k)$ are obtained by the expected number of times the model is in state S_j and emission V_k is observed, divided by the expected number of times the model is in state S_j. This gives

$$\bar{b}_j(k) = \frac{\sum_{t=1}^{L} \gamma_t^{O_t=k}(j)}{\sum_{t=1}^{L} \gamma_t(j)} \tag{6}$$

where $\gamma_t^{O_t=k}(j)$ represents the probability of the model residing in state S_j at time t given O, and the observed emission at time t is k.

B. The MA Algorithm

The MA algorithm, described in [13], differs from the BW algorithm and equivalent algorithms in two important ways. First, rather than completely re-estimating the model parameters at each step, the MA algorithm makes incremental updates to the model parameters. As a result, the parameter changes are smoother in the MA algorithm. This can mean, however, that more iterations are required to converge compared to the BW algorithm. The second, and perhaps most important, difference is that the MA algorithm uses observation sequences which belong to other classes (i.e. activities) than the modeled class, in addition to observation sequences belonging to the modeled class. In contrast, the BW algorithm only uses positive examples of observation sequences to determine model parameters. In other words, the MA algorithm uses both positive and negative examples to perform sequence discrimination.

The MA algorithm makes use of small, incremental parameter changes rather than re-estimation. As such, the update formulas for the MA algorithm are of a different nature than those used by the BW algorithm. Parameter updates are similar to those described in [14] (which describes a non-discriminative procedure), and are given by

$$\bar{a}_{ij} = \frac{e^{\lambda w_{ij}}}{\sum_{h=1}^{N} e^{\lambda w_{ih}}} \tag{7}$$

$$\bar{b}_j(k) = \frac{e^{\lambda v_j(k)}}{\sum_{h=1}^{M} e^{\lambda v_i(h)}} \tag{8}$$

where λ represents a (constant) learning rate value, w_{ij} is a $N \times N$ matrix of values representing the hidden state transition probabilities, and $v_j(k)$ is a $N \times M$ matrix of values representing the emission probabilities. In the MA algorithm, w_{ij} and $v_j(k)$ are updated, instead of updating a_{ij} and $b_j(k)$ directly. Each iteration, w_{ij} and $v_j(k)$ are updated according to $w_{ij} = w_{ij} + \Delta w_{ij}$ and $v_j(k) = v_j(k) + \Delta v_j(k)$.

As the MA algorithm makes use of observation sequences belonging to different classes for parameter re-estimation, O_s is defined as the s-th sequence in a set of observation sequences. Further, P_s is defined as the probability of observation sequence O_s being produced by the hidden Markov model, i.e. $P_s = P(O_s)$. For each observation sequence, a target probability P_s^* is defined. The MA algorithm then tries to minimize a distance metric with respect to P_s and P_s^*. The following parameters are defined:

$$d_s = \log(\frac{P_s^*}{P_s}) \tag{9}$$

$$d_{max} = \log(\frac{p_{max}^*}{P_{min}^*}) \tag{10}$$

where P^*_{min} is the minimum value of all P^*_s, and P^*_{max} the maximum value of all P^*_s. In practice, d_{max} can be set to any number provided $d_{max} > |d_s|$ for all s. Δw_{ij} and $\Delta v_j(k)$ are then given by

$$\Delta w_{ij} = C_a \sum_s \frac{d_s}{d^2_{max} - d^2_s} \sum_{t=1}^{L_s} (\varepsilon_t(i,j) - a_{ij}\gamma_t(i)) \quad (11)$$

$$\Delta v_j(k) = C_b \sum_s \frac{d_s}{d^2_{max} - d^2_s} \sum_{t=1}^{L_s} (\gamma_t^{O_{s,t}=k}(j) - b_j(k)\gamma_t(j)) \quad (12)$$

where C_a and C_b are constants. For most practical applications, C_a and C_b are set to 1. In the above equations, the first component ($\frac{d_s}{d^2_{max} - d^2_s}$) indicates whether to increase or decrease the likelihood of the model producing the given sequence O_s, depending on the current and target probabilities P_s and P^*_s. The second components in both equations represent the difference between the expected parameter probabilities and the current model parameter probabilities.

C. Coupled Hidden Markov Models

Coupled hidden Markov models (c-HMM), described in [15], in essence consist of a number of hidden Markov models linked together by introducing additional transition probabilities between hidden states of the different models. The amount of hidden Markov models making up a c-HMM is called the number of chains. Consider, for example, a two-chain c-HMM, consisting of two HMMs with N_x and N_y hidden states, respectively. The c-HMM can then be characterized by an $N_x + N_y \times N_x + N_y$ transition probability matrix. Here, the top-left quadrant and bottom-right quadrant represent the original transition probabilities of the regular HMMs. The remaining quadrants represent transition probabilities from the first HMM to the second and vice-versa. Note that both regular HMMs retain their original emission probabilities. A single emission probability matrix for the c-HMM can be obtained by taking a vector of emissions as input, and recomputing emission probabilities accordingly.

In a c-HMM, each chain resides in one of it's own hidden states, much as they would if they were uncoupled, regular HMMs. Also, at each time step, a separate emission is observed for each of the chains. The state of a c-HMM can therefore be expressed as the combination of states of all individual chains, and the emissions as a vector of all emissions of the individual chains. Indeed, a c-HMM can be fully expressed as the Cartesian product of the HMMs making up the chains.

As all c-HMMs described in the context of this document are two-chain models, they will be the focus in this section. However, the same principles apply to models of any number of chains. As there is both audio and video data available for activity recognition, a two-chain c-HMM allows modeling the audio process and the video process separately, rather than as a single entity. In other words, audio and video can be modeled as two independent, but interacting, processes.

In regular HMMs, the current hidden state is solely dependant on the previous hidden state, as stated by the Markov property. c-HMMs break the Markov property in this respect, as the current hidden state in a chain of a c-HMM is dependant on both the previous state in that chain, as well as the previous states of every other chain. Note that the current hidden state of a chain is not dependent on the current states of any other chains.

As there are now two or more concurrent hidden states to keep track of, the standard algorithm for evaluating a hidden Markov model (called the forward-backward procedure, see [12]) will no longer work, and cannot easily be adapted. Adaptations have been devised ([16]), generally at the loss of accuracy to reduce computational complexity to manageable levels. An alternative is to first transform the c-HMM into a regular, 'joint' HMM, and then use the standard forward-backward procedure. This transformation is accomplished by creating an HMM with the Cartesian product of the hidden states from each chain, where each hidden state of the joint HMM represents a combination of hidden states from the chains. The state transition probabilities are recomputed from the state transitions in the c-HMM. The resulting computational complexity is exponential in the number of chains. However, as only c-HMMs of two chains are used in this work, the computational costs remain manageable. If x and y represent individual chains and c represents the Cartesian HMM, the hidden state transition probabilities for the Cartesian HMM can be defined as

$$a_{c_{ip}c_{jr}} = P(q_{t+1} = S^c_{j,r} | q_t = S^c_{i,p}) \quad (13)$$

In addition, let

$$P^x = P(q^x_{t+1} = S^x_j | q^x_t = S^x_i, q^y_t = S^y_p) \quad (14)$$

$$P^y = P(q^y_{t+1} = S^y_r | q^x_t = S^x_i, q^y_t = S^y_p) \quad (15)$$

by means of which 13 can be rewritten as

$$a_{c_{ip}c_{jr}} = P^x \cdot P^y \quad (16)$$

As described above, transition probabilities depend on all current states of all chains. If independence between x and y is assumed, the components of the equation above can be written as

$$P^x = P(q^x_{t+1} = S^x_j | q^x_t = S^x_i) \cdot P(q^x_{t+1} = S^x_j | q^y_t = S^y_p) \quad (17)$$

$$P^y = P(q^y_{t+1} = S^y_r | q^x_t = S^x_i) \cdot P(q^y_{t+1} = S^y_r | q^y_t = S^y_p) \quad (18)$$

In practice, the assumption of independence may not hold. However, the above equations still provide a fairly accurate approximation in that case. Inserting these equations gives for the Cartesian hidden state transition probabilities

$$a_{c_{ip}c_{jr}} = a_{x_i x_j} a_{y_p y_r} a_{x_i y_r} a_{y_p x_j} \quad (19)$$

Similar reasoning can be applied to the Cartesian emission probabilities, giving

$$b_{c_{ip}}(k) = b_{x_i}(k) b_{y_p}(k) \tag{20}$$

The parameter estimation algorithms, such as BW, can then be used mostly unaltered. The only issue arises when the parameters for the c-HMM need to be re-estimated. Whilst it is fairly straightforward to determine the joint HMM's parameters from the c-HMM's parameters, the opposite is not true, introducing some amount of error in the re-estimation. The algorithm used in this work is based on the method described in [15], by factoring after re-estimation of the joint HMM. Theoretically, convergence to a local optimum is not guaranteed. However, in practice this has not presented any problems. As described in [15], convergence can be guaranteed by applying a gradient descent algorithm after re-estimation.

IV. Experimental setup

To both test and train an activity recognition system, a set of data labeled with ground truth activity values must be created. To obtain this data, a number of recordings have been made in Philips' ExperienceLab[1] in Eindhoven, a facility specifically designed for performing experiments and creating recordings in a home environment. Participants were invited to individually perform six selected activities.

The experiment was set up in the kitchen area of the ExperienceLab, using a single camera mounted on the ceiling in one of the corners of the kitchen, and a single microphone mounted centrally on the ceiling. After the participants arrived, they were given a short tour of the kitchen, explaining the location of any appliances or utensils they might need, and the operation of some of the appliances such as the stove.

After they were comfortable around the kitchen, they were given a list detailing the six activities to perform: storing a set of groceries from a bag they were given at the start of the experiment, preparing a meal, eating the meal, doing the dishes, vacuuming the kitchen floor and preparing a drink. The participants were asked in advance which meal they would like to prepare and eat, so ingredients could be provided. Participants were allowed to either do the dishes by hand or use the dishwasher in the kitchen. They were allowed to perform the activities in any order, in whatever manner they wished. The only exception was the request to eat at a specified table, so the participants would remain within the camera's view. Most participants completed the activities within a time span of 30 minutes to an hour.

Afterwards, the recorded sessions were annotated by hand to include the ground truth activities, and audio and video events. Table I shows the list of annotated classes. The activity classes were chosen such that they contained considerable overlap of observed events, and therefore could not be distinguished purely by the occurence of one or more events. The exception is the activity 'Vacuuming', which can generally be identified by the sound of a vacuum cleaner. Unfortunately, annotating data is expensive and time-consuming, and as such unlikely to be available in practice. An alternative is the use of video and audio scene analysis techniques to automatically infer relevant events from the raw video and audio signals. A downside of using such methods is that the annotations created by these algorithms are likely to be less accurate than annotations created manually.

Table I
THE LIST OF CLASSES FOR EACH ANNOTATION TRACK. THE 'ACTIVITIES' TRACK LISTS PARTICIPANT ACTIVITIES, 'AUDIO EVENTS' ARE RELATED TO OBSERVED SOUNDS, AND 'VIDEO EVENTS' ARE RELATED TO THE POSITION OF A PARTICIPANT IN THE KITCHEN.

Activities	*Audio events*	*Video events*
Storing groceries	Background	Fridge
Preparing dinner	Groceries bag	Dishwasher
Eating	Kitchen door	Sink
Doing dishes	Groceries	Cupboard
Vacuuming	Fridge	Stove
Preparing a drink	Fridge door	Dining table
Other	Pans	Transition
	Cutlery	Other
	Tap	
	Stove on /off	
	Stove	
	Plates	
	Chair	
	Movement	
	Glass	
	Hot water	
	Vacuum cleaner	
	Cleaning the counter	
	Dishwasher	
	Pouring a drink	
	Voice	

The scene analysis algorithms used in this work include an audio classification algorithm based on Gaussian mixture models, which works on a set of frequency domain features extracted from the raw audio signal. Gaussian mixture models is a popular algorithm in audio scene analysis, for example [17]. The algorithm uses part of the manually annotated audio data both for training and testing. The video classification algorithm is based on a background subtraction model, using manually defined regions of interest in the kitchen. The algorithm uses part of the manually annotated data for testing purposes. The outputs of both algorithms correspond to categories defined in the audio and video tracks of the annotated data. Cross validation results on the annotated data have shown that the performance of the audio scene analysis algorithm is in the region of $60 - 70\%$, depending on the evaluated recorded session. The performance of the video scene analysis algorithm is approximately 71%.

As some activities occur more frequently in the data than

[1] See http://www.research.philips.com/focused/experiencelab.html

Table II
SEQUENCE-BASED ACTIVITY CLASSIFICATION ACCURACY RATES ON ANNOTATED DATA AND SCENE ANALYSIS DATA USING COMPLETE EVENT SEQUENCES. THE TABLES LIST THE ACCURACIES USING THE BAUM-WELCH ALGORITHM (BW), THE MA ALGORITHM, THE COUPLED BAUM-WELCH ALGORITHM (CBW) AND THE COUPLED MA ALGORITHM (CMA).

	BW	*MA*	*cBW*	*cMA*
Groceries	1	1	1	0.63
Dinner	0.94	0.94	0.94	0.88
Eating	0.89	1	0.89	1
Dishes	1	1	0.91	0.91
Vacuuming	1	1	1	1
Drinks	0.79	0.86	0.71	0.79
Class accuracy	0.94	0.97	0.91	0.87

(a) Annotated data

	BW	*MA*	*cBW*	*cMA*
Groceries	1	1	0.88	1
Dinner	1	1	1	0.94
Eating	0.67	0.89	0.89	0.89
Dishes	1	1	1	1
Vacuuming	1	1	1	1
Drinks	0.43	0.21	0.21	0.79
Class accuracy	0.85	0.85	0.83	0.94

(b) Classified data

others (for example, some participants prepared a drink multiple times during their session), using the standard measure of accuracy (number of correctly classified instances divided by the total number of classified instances) results in a measure biased by the accuracy on the most common class(es) in the dataset. Therefore, class accuracy is used instead. Here, standard accuracy is computed for each activity class individually, and class accuracy is defined as the mean of the standard accuracies for all classes. From this point on, the terms 'accuracy' and 'class accuracy' will be used interchangeably.

V. RESULTS

A. Sequence-based classification

From the recorded data, a set of event sequences can be generated for each activity. For a given recorded activity, the matching event sequence is given by the sequence of events starting at the activity's start time and ending at the activity's ending time. The generated sets of event sequences can then be used in training and evaluating the activity models. This provides a solid indication of the classifying power of the models, albeit not an overly realistic one, as the start and end times of activities will in practice often be unclear.

The resulting accuracy scores for the annotated data are shown in Table IIa. The described results were obtained using 10-fold cross validation. When considering the class accuracy scores, it can be seen that all methods perform well on classification of event sequences. This is especially true for the non-coupled classification methods, and for the MA method in particular, with a classification class accuracy of 97%. A slightly different picture can be seen when considering the results of sequence-based activity classification on the scene analysis data, shown in Table IIb. For most classification methods, class accuracy scores are lower compared to the annotated data. This is to be expected, as the errors made by the automatic classification of observations in the scene analysis data introduces additional noise to the dataset. Remarkably, the coupled MA method breaks this trend, and is considerably more accurate on the scene analysis data than on the annotated data. This is mainly due to the relatively high accuracy on the 'preparing a drink' activity, which has a considerably shorter average duration compared to the other activities examined.

B. Session-based classification

To simulate an application where in the moment classification of activities is required, classification is performed on the entire data stream of individual recording sessions. In such an application, there is no indication when one activity ends and another begins, nor is the entire event sequence of the current activity available. When activity recognition is being performed on one of the eight recorded sessions, the remaining seven sessions are used to train the activity recognition models (i.e. estimate model parameters), resulting in an analysis method similar to cross validation. The activity recognition algorithm uses a sliding window of a fixed length to store incoming data read from the audio and video streams. Whenever the contents of the window are updated by two events, activity recognition is performed on the current contents of the sliding window.

Whenever activity recognition is performed, the result is compared with the ground truth which has been annotated. Accuracy is defined as the fraction of matches between the recognized activity and the ground truth over all windows in the session. Windows for which the ground truth activity is 'other' are discarded for this purpose, as the recognition algorithm cannot recognize this class of activity. The average accuracy over all sessions is computed to determine the overall accuracy of the activity recognition algorithm.

A disadvantage of using a (large) window of events is that it introduces a 'delay' between the ground truth and the recognized activity; where a change in the ground truth only results in a change in the recognized activity several windows later. The cause of this is that when the ground truth changes, there will be at most a few observations in the current window which represent the new activity. The remaining observations related to the previous activity still remain, causing events produced by the previous activity to dominate events produced by the current activity. As the sliding window is updated, more observations of the current activity enter the window, while the older observations are pushed out, until eventually events produced by the current activity become dominant. This can also be seen in Figure 2, which shows an example activity recognition result.

Table III
ACTIVITY CLASSIFICATION CLASS ACCURACY RATES ON ANNOTATED AND SCENE ANALYSIS DATA USING AUDIO AND VIDEO-BASED CLASSIFICATION. THE TABLES SHOW THE ACCURACIES PER RECORDED SESSION, NUMBERED 1 THROUGH 8, AND THE AVERAGE ACCURACIES OVER ALL EIGHT SESSIONS. THE TABLES SHOW ACCURACY RESULTS FOR THE BAUM-WELCH ALGORITHM (BW), THE MA ALGORITHM, THE COUPLED BAUM-WELCH ALGORITHM (CBW) AND THE COUPLED MA ALGORITHM (CMA). FOR EACH SESSION, THE MOST ACCURATE CLASSIFICATION IS HIGHLIGHTED.

	1	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*acc*
BW	0.70	**0.83**	**0.86**	**0.75**	0.82	**0.87**	0.74	**0.74**	**0.79**
MA	0.65	0.81	0.83	0.70	0.83	0.76	0.76	0.64	**0.75**
cBW	**0.77**	0.79	0.82	0.73	0.81	0.76	0.73	0.72	**0.77**
cMA	0.74	0.77	0.79	0.70	**0.84**	0.78	**0.79**	0.68	**0.76**

(a) Accuracy on annotated data

	1	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*acc*
BW	0.65	0.60	0.64	0.66	0.55	0.69	**0.76**	0.65	**0.65**
MA	0.72	0.62	0.75	0.64	0.52	0.75	0.66	0.65	**0.66**
cBW	0.74	**0.65**	0.61	0.64	0.58	0.65	0.70	0.61	**0.65**
cMA	**0.80**	0.61	**0.82**	**0.71**	**0.61**	**0.82**	0.76	**0.71**	**0.73**

(b) Accuracy on scene analysis data

Table IV
CLASS ACCURACY ON RECORDED SESSIONS USING BOTH THE AUDIO AND VIDEO MODALITIES (A&V), AUDIO ONLY (A) AND VIDEO ONLY (V). ACCURACIES ARE SHOWN FOR THE BAUM-WELCH ALGORITHM (BW), THE MA ALGORITHM, THE COUPLED BAUM-WELCH ALGORITHM (CBW) AND THE COUPLED MA ALGORITHM (CMA).

	a&v	a	v
BW	0.79	0.78	0.56
MA	0.75	0.72	0.51
cBW	0.77	-	-
cMA	0.76	-	-

(a) Annotated data

	a&v	a	v
BW	0.65	0.55	0.50
MA	0.66	0.48	0.53
cBW	0.65	-	-
cMA	0.73	-	-

(b) Classified data

As a result, a larger sliding window size results in more errors due to the aforementioned delay. However, when the ground truth activity is stable fewer errors are made, as there is more information in the sliding window for the activity recognition algorithm to draw on. Therefore, an optimal compromise must be found: if the window size is too small, it contains too little information to base recognition on. If too large, changes in the ground truth will be detected too late or not at all. In either case, the overall error will likely become larger. Experimentation shows that a window size of 50 yields a good compromise between both of these pitfalls. This window size corresponds roughly to a duration of 25 seconds.

Making use of the methods described above, the result of the accuracies over each session and averages over all eight sessions using the annotated data and scene analysis data is shown in Tables IIIa and IIIb respectively. From the results, it is clear that classification on the annotated data yields considerably higher class accuracy scores compared to classification on the scene analysis data. This is to be expected, as the annotated data contains fewer errors compared to the scene analysis data. On the annotated data, all algorithms perform comparably, with the BW algorithm performing slightly above the rest. On the scene analysis data, the coupled MA algorithm outperforms the other algorithms, which perform at a similar level. A possible explanation for the performance of the coupled MA algorithm is a greater robustness against the noise introduced by the scene analysis data.

When considering the accuracy scores of the individual sessions, it can be seen there are considerable differences between sessions, and also between methods for the same session. However, some sessions appear to elicit higher accuracy scores than others, regardless of methods used. A good example of this is session 3. It therefore appears that some sessions are inherently more difficult to classify than others, likely due to individual differences between participants.

C. Effect of multiple modalities

To investigate whether the use of multiple modalities, in this case audio and video, results in improved performance compared to the use of only a single modality, similar experiments as in Section V-B were performed, this time using events from a single modality only. As new events are now less frequent, the size of the sliding window was reduced to 25 events, which corresponds to roughly 25 seconds of observations as before. In addition, the window contents are evaluated whenever a new event is detected, rather than every two events.

The average accuracies obtained using both modalities, as well as using each modality individually is shown in Table IV for both the annotated and scene analysis data. Note that the coupled methods are omitted for single modalities, as the use of a single chain would be equivalent to using the appropriate non-coupled method. It can be seen that for the annotated data, using audio as a single modality yields comparable results to using both modalities. However, on scene analysis data, using audio events only results

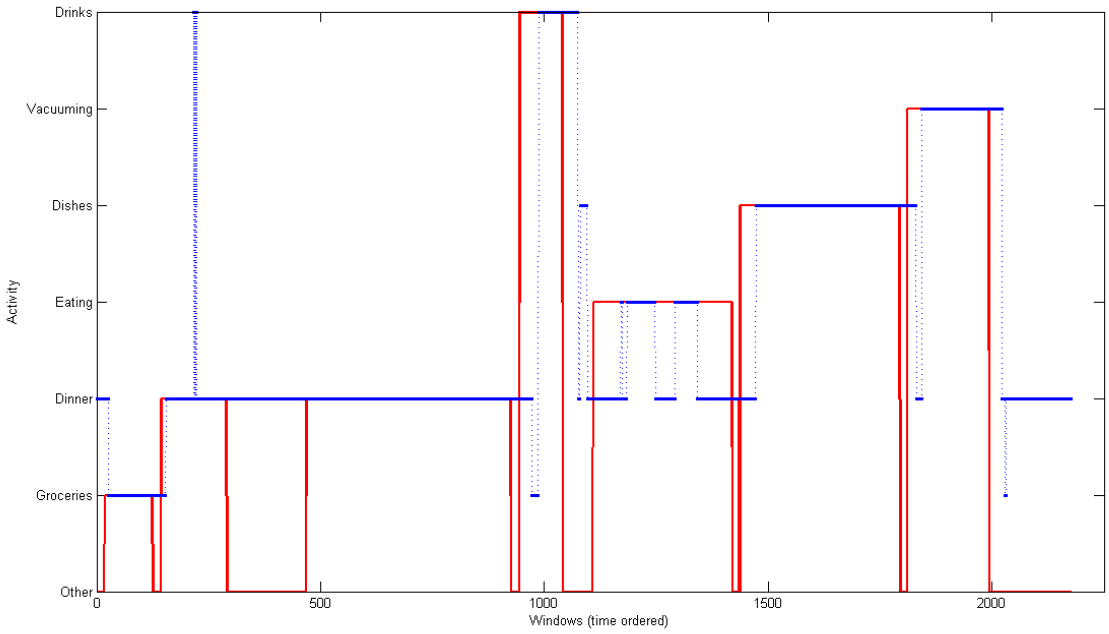

Figure 2. Example of activity recognition results on a recorded session. The red line in the graph indicates the ground truth activity at each time step during the recording, while the blue line indicates the recognized activity at each of those time steps. An overlap between the two lines indicates an accurate recognition, a difference indicates a recognition error. This example shows the results on session 1, using the annotated data and the Baum-Welch algorithm.

in considerably poorer classification accuracy, presumably affected by the noise in the dataset. Using video as single modality consistently yields inferior accuracy compared to using both modalities.

VI. Discussion

From the first experiment (Section V-A), it can be deduced that the constructed activity models can accurately distinguish between activities if the full event sequence is available. For some applications, classifying activities after their completion is acceptable, for example in the case of monitoring independently living elderly. In practice, determining activity start and end times may be a challenge. In some cases, a lack of sensor activity may indicate an activity has started or ended.

The second experiment (Section V-B) shows lower accuracy rates, as is expected as less knowledge is available to the classifier. On the annotated data, the highest overall classification accuracy using both audio and video is approximately 79%, the highest overall classification on the scene analysis data approximately 73%. For both data sets, the classification results indicate that the majority of user activities can be accurately recognized across participants, also when considering the effects of ground truth / recognized activity delays (see Section V) on the classification accuracy. Also, individual differences in classification accuracy can be seen between participants. As classification models were trained using data of other participants, it seems likely that participants with relatively high accuracy scores performed their activities more conform to some general model of activity. To properly explain the difference in accuracy between participants however, more research would be required. In both the first and second experiment, the Baum-Welch algorithm yields better accuracy on the annotated data, while the coupled MA algorithm yields better accuracy on the scene analysis data.

For the third experiment (Section V-C), it can be concluded that the combination of both audio and video modalities yields a considerable improvement in classification on scene analysis data. On the relatively less noisy annotated data, audio-only classification yields very similar results to multi-modal classification. It therefore seems that given the availability of highly accurate training data, audio information is sufficient to accurately classify users' activities. In practice, however, such data is unlikely to become available unless additional steps in audio and video scene analysis are made.

There are a number of limitations to the methods proposed in this work. First, activities for which a model is not available cannot be classified, or even identified as being different from the modeled activities. Several solutions can be conceived of to deal with this problem, including

the construction one or more activity supermodels, each representing a subset of activities not modeled explicitly. Unfortunately, the recordings contained insufficient data to experiment with this. An alternative is the use of confidence estimation techniques to identify situations where none of the activity models provide a good match for the current observations. For example, early results indicate that a neural network with the model likelihoods as input can identify a significant number of errors with little loss of accuracy.

A second limitation is that the scene analysis algorithms currently require retraining when moved to a new environment or are likely to suffer reduced accuracy. The authors are currently investigating methods to adapt these algorithms to new environments in an unsupervised manner, for example by autonomously identifying common events, removing the need for new manually labeled training data. Whether such methods prove feasible is currently under investigation.

The results described in the previous section also show that the proposed methods are able to overcome a number of common challenges associated with ADL classification. First, the results show that the proposed methods are comparible with methods in which a greater number of sensors has been used, or with methods which make use of more invasive sensor modalities. Secondly, the results show that the methods proposed are robust across participants, eliminating the need for retraining when a new user is encountered.

VII. Conclusion

In this document, an approach has been presented to classify ADLs using a single unit containing a camera and microphone. It has been shown that this approach creates activity models which are generalizable across different users, eliminating the need for retraining when a new user is encountered. A number of techniques for constructing activity models have been outlined and tested experimentally. If activities are observed entirely, a classification accuracy of 97% and 94% can be obtained on annotated and scene analysis data respectively. When using an event history of up to 25 seconds, an accuracy can be achieved of 79% on annotated data, and an accuracy of 73% on scene analysis data. On the former the Baum-Welch algorithm performs slightly better than the other methods, while on the scene analysis data the more complex coupled MA algorithm yields the highest accuracy. Further, it was shown that using a combination of audio and video events as input yields superior results compared to using either modality individually when considering scene analysis data.

References

[1] M. Philipose, K. P. Fishkin, M. Perkowitz, D. J. Patterson, D. Fox, H. Klautz and D. Hahnel, *Inferring Activities from Interactions with Objects*, IEEE Pervasive Computing, 3:4, 2004

[2] P. Rashidi and D. J. Cook, *Keeping the Resident in the Loop: Adapting the Smart Home to the User*, IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans, 39:5, 2009

[3] T. van Kasteren, A. Noulas, G. Englebienne and B. Krose, *Accurate activity recognition in a home setting*, Proceedings on the 10th international conference on Ubiquitous computing, 2008

[4] L. Bao and S. S. Intille, *Activity recognition from user-annotated acceleration data*, Lecture Notes in Computer Science, Springer, 2004

[5] J. Lui, Z. Wang, L. Zhong, J. Wickramasuriya and V. Vasudevan, *uWave: Accelerometer-based personalized gesture recognition and its applications*, IEEE International Conference on Pervasive Computing and Communications, 2009

[6] P. Turaga, R. Chellappa, V. S. Subrahmanian and O. Udrea, *Machine Recognition of Human Activities: A Survey*, IEEE Transactions on Circuits and Systems for Video Technology, 18:11, 2008

[7] T. Chin, L. Wang, K. Schindler and D. Suter, *Extrapolating Learned Manifolds for Human Activity Recognition*, IEEE International Conference on Image Processing, 1, 2007

[8] R. Bodor, R. Morlok and N. Papanikolopoulos, *Dual-camera system for multi-level activity recognition*, IEEE International Conference on Intelligent Robots and Systems, 1, 2004

[9] C. Wojek, K. Nickel and R. Stiefhagen, *Activity Recognition and Room-Level Tracking in an Office Environment*, IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems, 2006

[10] L. N. Abdullah and S. Noah, *Integrating Audio Visual Data for Human Action Detection*, Fifth International Conference on Computer Graphics, Imaging and Visualization, 2008

[11] N. Oliver, E. Horvitz and A. Garg, *Layered Representations for Human Activity Recognition*, Fourth IEEE International Conference on Multimodal Interfaces, 2002

[12] L. R. Rabiner, *A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition*, Proceedings of the IEEE 77 (2), 1989

[13] H. Mamitsuka, *Supervised Learning of Hidden Markov Models for Sequence Discrimination*, Annual Conference on Research in Computational Molecular Biology, 1997

[14] P. Baldi and Y. Chauvin, *Smooth On-line Learning Algorithms for Hidden Markov Models*, Neural Computing, 6, 1994

[15] M. Brand, N. Oliver and A. Pentland, *Coupled Hidden Markov Models for Complex Action Recognition*, Computer Vision and Pattern Recognition, 1997

[16] M. Brand, *Coupled Hidden Markov Models for Modeling Interactive Processes*, MIT Media Lab Perceptual Computing / Learning and Common Sense Technical Report 405, 1997

[17] D. A. Reynolds, T. F. Quatieri and R. B. Dunn, *Speaker Verification Using Adapted Gaussian Mixture Models*, Digital Signal Processing, 10:1-3, 2000