Object-centred recognition of human activity

Nabiei, Roozbeh; Parekh, Manish; Jean-Baptiste, Emilie; Jancovic, Peter; Russell, Martin

DOI:
10.1109/ICHI.2015.14

Document Version
Early version, also known as pre-print

Citation for published version (Harvard):
Nabiei, R, Parekh, M, Jean-Baptiste, E, Jancovic, P & Russell, M 2015, 'Object-centred recognition of human activity', Paper presented at IEEE International Conference on Healthcare Informatics, Dallas, Texas, United States, 21/10/15 - 23/10/15. https://doi.org/10.1109/ICHI.2015.14

Object-centred recognition of human activity

Roozbeh Nabiei, Manish Parekh, Emilie Jean-Baptiste, Peter Jančovič and Martin Russell
School of Electrical and Systems Engineering, University of Birmingham, Birmingham, B15 2TT, UK
Email: [rxn946, m.parekh, emj198, p.jancovic, m.j.russell]@bham.ac.uk

***Abstract*—This paper describes an approach to real-time human activity recognition using hidden Markov models (HMMs) and sensorised objects, and its application to rehabilitation of stroke patients with apraxia or action disorganisation syndrome (AADS). Results are presented for the task of making a cup of tea. Unlike speech or other sequential decoding problems where HMMs have previously been successfully applied, human actions can occur simultaneously or at least in overlapping time. The solution proposed in this paper is based on a parallel, asynchronous set of detectors, each responsible for the detection of one of the component sub-goals of the tea-making task. The inputs to these detectors are formed from the outputs of sensors attached to the objects involved in that sub-goal, plus hand coordinate data. The sensors, comprising an accelerometer and three force-sensitive resistors, are packaged in a coaster which can be easily attached to the base of a mug or jug. In tests on complete tea-making trials, error rates range from less than 5% for sub-goals where all of the objects involved are sensorised, to up to 30% for detectors that rely on hand-coordinate data alone. The complete set of detectors runs in real-time. It is concluded that a set of parallel HMM-based sub-goal detectors combined with fully sensorised objects, is a viable, accurate and easily deployable approach to real-time object-centred human activity recognition.**

I. Introduction

In the UK alone it is estimated that over 150,000 people have a stroke each year [26]. Approximately 68% of survivors suffer from Apraxia or Action Disorganization Syndrome (AADS), leading to an impairment of cognitive abilities to complete activities of daily living (ADLs) [3], [5]. For example, patients might perform a wrong sequence of actions, skip steps, or misuse objects with possible safety implications. Caregivers can provide assistance, but patients who aspire to independent living may be unwilling to accept this as a long-term solution. Hence, the objective of the CogWatch project [11], [27] is to develop an intelligent computer-based rehabilitation system to re-train patients to carry out ADLs. To achieve this the system must be able to monitor the patient's progress through the ADL and provide appropriate guiding cues or feedback when an error is detected or anticipated. Recognising the individual actions that make up the ADL, and planning the patient's optimal strategy during the task are critical. In CogWatch, planning is achieved using a task model based on a Markov decision process (MDP) or a partially observable MDP (POMDP) [15]. This paper presents on a solution to the Action Recognition (AR) problem.

The initial ADL in CogWatch is "making a cup of tea". Task analysis [2] is applied to represent tea-making as a hierarchical 'task' tree, with sub-goals such as "Fill Kettle" (using water from a pre-filled jug) and "Add Teabag" (the complete list of sub-goals is given in section II).

These sub-goals are recognised from the outputs of sensors attached to the objects involved, and the location of the hands. Variations in the sequences of sensor outputs that result from individual differences in the ways that users execute the task, variations in the way that the same user executes the same task on different occassions, or sensor noise are captured using a statistical model (a sub-goal hidden Markov model (HMM) (for example, [20])). The partially-ordered structure of the sub-goal lattice, in which sub-goals occur in overlapping time, or even at the same time, is accommodated using a parallel set of asynchronous HMM-based detectors, each responsible for detecting a specific sub-goal.

Capturing an ADL through sensors [1], [10], [12], [13], [16], [18], [23], [8], or using HMMs to recognize human activity [17] are certainly not new, but decomposition of an ADL into sub-goals and recognition of these sub-goals has received less attention.

The use of sensorised objects promotes an "object-centric" view of action recognition, in which a sub-goal is characterised in terms of how it is "experienced" by the objects involved. This contrasts with "scene-oriented" approaches, in which an external video sensor plus image processing is used to identify and track the hands and objects during a task, or approaches where sensors are attached to the body (for example [22], [23]). The object-centred and scene-oriented approaches are both unobtrusive, since neither requires the user to wear sensors. However, the scene-oriented approach normally requires careful installation and calibration of cameras, which may be an issue if the system is intended to be widely deployed and stand-alone, for example in an ordinary household kitchen.

A popular option for instrumentation is to use Radio Frequency Identification (RFID) tags to identify which objects have been picked up [4], [24], however these do not provide sufficiently rich information and an antenna needs to be worn.

Although the objective of CogWatch is the wider development of technology for cognitive rehabilitation of stroke patients, the focus of the present paper is the development of the required action recognition system, based on HMMs and instrumented objects. The Task Model, which uses the outputs of the action recognition system to monitor the patient's progress through the task and detect errors, is described elsewhere [15]. The paper is organised as follows. Section II describes the task. Section III describes our approach to instrumentation of objects. Section IV describes how features are extracted from the sensors. Sections V and VI describe the action recognition system, and sections VII and VIII present an experimental evaluation of the system. A discussion of results is presented in Section IX. Section X presents our conclusions.

Fig. 1. *A jug fitted with a CogWatch Instrumented Coaster (CIC) and an 'open' CIC, showing the accelerometer, PIC, Bluetooth module and battery*

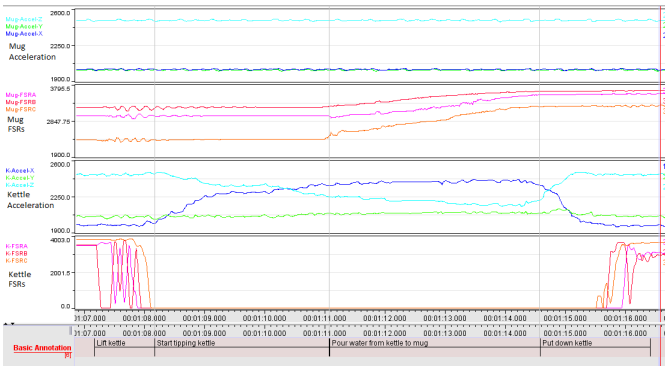

Fig. 2. *Outputs from the mug (top two graphs) and kettle (bottom two graphs) during an execution of the "Pour kettle" sub-goal.*

II. Analysis of Tea-making

CogWatch provides assistance for four types of tea-making "black tea", "black tea with sugar", "tea with milk" and "tea with milk and sugar". Using task analysis [14], each variant is decomposed into a hierarchy of sub-goals, tasks and sub-tasks. At the first level, eight sub-goals were identified, plus a common error (9) and a potential hazard (10). These are:

1) "Fill Kettle" (using water from a pre-filled jug)
2) "Boil Water"
3) "Pour Kettle" (i.e. pour boiling water into the mug)
4) "Add Tea-bag"
5) "Add Sugar"
6) "Add Milk"
7) "Remove Tea-bag"
8) "Stir"
9) "Toy Milk" (pour milk outside the mug)
10) "Toy Kettle" (pour boiled water outside the mug)

This list is a high-level description of the sub-goals of tea-making. It is not a prescription for a linear sequence. The execution of sub-goals may overlap, so that one sub-goal begins before another is complete (for example, the user may execute several sub-goals during "Boil Water" , or if both hands are used "Add tea-bag" could start during "Pour kettle"). Even when the sub-goals do occur in sequence the order may vary. Hence a particular instance of tea-making is more accurately represented as a partially-ordered lattice of sub-goals. This complicates AR (Section V) and makes it difficult to use sequential information, for example in the form of a syntax or grammar, to improve recognition accuracy.

III. Instrumentation and Sensors

A. The CogWatch Instrumented Coaster (CIC)

The objects involved in the tea-making task are a kettle, water jug, mug, milk jug, spoon and containers for the tea-bags, sugar and used tea-bags. In the current system only the kettle, mug and milk jug are instrumented. To avoid patient confusion, the objects must appear normal and function as expected. Hence the sensors need to be small and discrete. The chosen solution is to package the sensors and circuitry into an instrumented 'coaster', the 'CogWatch Instrumented Coaster (CIC)', that is fitted to the underside of the object (figure 1). This is inspired by the MediaCup concept [8]. The CIC contains a 3-axis accelerometer, 3 force sensitive resistors (FSRs), a PIC, a Bluetooth and a battery. For the kettle, which is 'cordless' with a separate base, the CIC was split into two packages, with the accelerometer attached to the kettle body and the FSRs attached to the base. The accelerometer is an Analog Devices ADXL335, providing acceleration measurements on 3 axes in a range of ± 3g. Its function is to respond to changes in motion, tilting, and disturbances of the object due to the addition of materials, stirring, collisions or (in the kettle) vibration during boiling. The FSRs can detect whether the object is standing on a surface of lifted in the air, changes in weight due to the addition or removal of materials, and more subtle changes in weight distribution across the base of the object (making it possible, for example, to detect stirring).

Figure 2 shows example outputs from the CICs attached to the mug (top two graphs) and kettle (bottom two graphs) during an execution of the "Pour kettle" sub-goal. The output of an individual CIC at any time is a six dimensional vector, comprising x, y, z accelerometer outputs plus the outputs of the three FSRs. The data from the FSRs attached to the mug (second graph from top) show the increase in weight of the mug as it is filled. The data from the kettle FSRs (bottom graph) clearly show the points where the kettle is lifted from and then returned to the table

B. Kinect-based Hand-tracking

In addition to outputs from CICs, the system uses hand-coordinate data captured using Kinect [6], using software based on the 'Kinect-Arms' libraries [9].

IV. Feature Extraction

The raw data (comprising hand coordinates from Kinect, and FSR and accelerometer data from the three CICs) are streamed to the system and synchronised at 50 Hz. Each sub-goal is characterised by a different combination of raw sensor data and features extracted from the raw sensor data. For example, detection of the sub-goal "Pour Kettle" uses the outputs from the kettle CIC, the FSRs in the CIC attached to the mug, and hand position.

Hand position is given relative to x and y axes parallel to edges of the table and centred at the centre of the table. A 2D "Gaussian neighbourhood" associated with each object, is used to indicate when the hand is in the vicinity of that object. The mean and covariances of the Gaussian neighbourhood for an object is calculated using the location of the hand when it is stationary and interacting with that object. The hand is assumed to be stationary if the difference between successive

samples is less than 3mm. The distance that the hand has travelled between times t and $t+1$ is the Euclidean distance:

$$d(h_t, h_{t+1}) = \sqrt{(h_{1,t+1} - h_{1,t})^2 + (h_{2,t+1} - h_{2,t})^2}.$$

Here $h_t = (h_{1,t}, h_{2,t})$ is the position of the hand at time t.

A number of features are extracted from the raw data for AR. For example, to calculate the change in weight of the mug a low pass filter is used to smooth the data from FSRs in the CIC attached to the mug, before the derivative is calculated. Also, the FSR data obtained from the FSRs under the kettle and in the CIC attached to the milk jug is used to determine whether or not that object has been picked up. Variance in the energy of the outputs from the accelerometer attached to the kettle body, caused by vibration of the kettle during the process of heating the water, is used to determine whether the water in the kettle had reached boiling point and hence detect the sub-goal "Boil Water".

The feature vector y_t at time t is calculated from a window comprising sensor outputs at times $t-20, ..., t$.

V. HMM-Based Action Recognition

Hidden Markov models (HMMs) are a generic framework for statistical sequential pattern processing, but they have received most attention in the area of automatic speech recognition (ASR) (for example, see [7]). However, there are a number of important differences between AR and ASR which determine the design of our HMM-based AR system:

- In ASR words occur one-after-another, whereas in AR actions can occur in overlapping time, so that the natural structure is a partially-ordered lattice rather than a sequence. Overlap may occur, for example, if the subject uses both hands, or executes one or more sub-goals while the kettle is boiling. Therefore a conventional ASR decoder, which will compute the most probable sequence of actions, is not appropriate for AR. This partially-ordered structure also complicates the inclusion of sequential constraints in the decoder.
- In ASR the same features are used by all HMMs, whereas in AR different subsets of features are appropriate for recognising different sub-goals.
- In AR there is no accepted equivalent to a 'phone set'.

The key process in a typical HMM-based ASR system is a Viterbi decoder [7]. Given a sequence of feature vectors $y = y_1, ..., y_T$ the Viterbi decoder finds the sequence of HMMs $M = M_1, ..., M_N$ such that an approximation to the probability $p(M|y)$ is maximised. Since y is fixed, from Bayes' rule this is equivalent to finding M such that $p(y|M)P(M)$ is maximised. The probability $P(M)$ is based on a language model which defines the probability of any given sequence of words. In speech recognition, the language model and the individual HMMs are compiled into a single network and the most probable path through this network is found using Viterbi decoding. However, because the execution of sub-goals is realised as a partially-ordered lattice, an alternative architecture is needed for AR.

VI. Detector structure

1) Detector architecture: The AR system comprises five independent real-time HMM-based detectors which together can identifying occurrences of the eight sub-goals of tea-making at any time during completion of the task. These detectors run in parallel and are mutually independent. Each detector takes as input those parts of the feature vector that are useful for detecting its sub-goal(s). A detector consists of one or more multiple state HMMs, each representing a unique sub-goal, and these HMM states are associated with Gaussian mixture models (GMMs). In addition, the detector includes a single state "background" (or "toying") HMM, whose state is associated with a multiple-component GMM.

2) Viterbi decoding: An identical implementation of the Viterbi algorithm (for example see [7]) runs independently in each decoder. Briefly, each detector works as follows: At each time t the detector receives a new feature vector, y_t. For each state i of each of its HMMs, a quantity $\alpha_t(i)$ is calculated which can be thought of as an approximation to the probability of the best explanation of data $y_1, ..., y_t$ up to and including y_t ending in state i at time t. Intuitively, if the detector is for "Add Milk" and the i^{th} state corresponds to tipping the jug, then $\alpha_t(i)$ can be thought of as the probability of the best explanation of data up to time t culminating in the tipping action at t. Formally $\alpha_t(i)$ is given by the recursion:

$$\alpha_t(i) = max_j \alpha_{t-1}(j) a_{j,i} b_i(y_t) \quad (1)$$

$$\rho_t(i) = argmax_j \rho_{t-1}(j) a_{j,i} b_i(y_t) \quad (2)$$

where $a_{j,i}$ is the probability of a transition from state j to state i and $b_i(y_t)$ is the probability of the sensor data y_t given state i. Note that the 'preceding' state j can be in the same HMM as state i, or, if i is an initial state, j can be the final state of another HMM in the detector. $\rho_t(i)$ provides a record from which the best explanation of the data up to time t in state i can be recovered.

3) Partial traceback: In the basic implementation of Viterbi decoding described above, the best explanation of the data is not recovered until the final time T. However, in a real-time implementation there is no final time. The memory required to store the $\rho_t(i)$s and $\alpha_t(i)$s will increase and no output will be produced. The solution is to use a technique called 'partial traceback' [21]. Each detector's output up to a time s is generated as soon as its classification of the data up to that point is unambiguous, in the sense that all of the $\rho_t(i)$s can be traced-back to a common state at time s in the past. The memory used to store alternative explanations of the data up to s is then freed. In this way the decoders can run indefinitely. If the convergence point s is significantly less than t then there will be a delay in the output of the decoder. Therefore, care is needed in the construction of the HMMs to avoid the ambiguity that will cause this to happen.

Whenever a sub-goal HMM provides the most probably explanation of a section of input, a label indicating that sub-goal is output. Otherwise the best explanation of the data is "toying" and nothing is output.

4) Detector structure: The five detectors are as follows:

- The "Front Actions" detector consists of three "sub-goal" models (corresponding to "Add Sugar", "Add

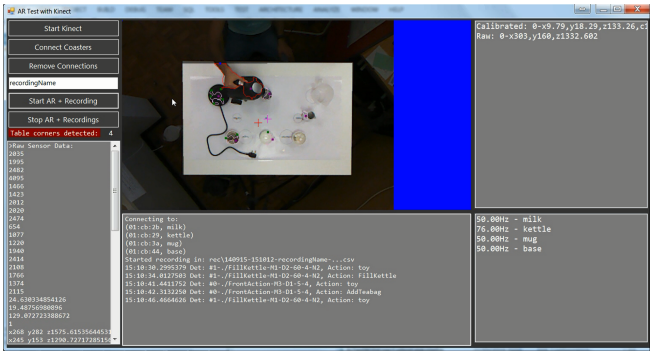

Fig. 3. *Screen shot showing the output of the real-time action recognition system.*

Tea-bag" and "Remove Tea-bag") and a background "toying" model. This detector is primarily influenced by the Gaussian neighbourhood features for the mug, tea-bag container, sugar container and used tea-bag container, which are calculated from Kinect, and the outputs of the FSRs in the CICs under the mug (see section IV).

- The "Pour Kettle' and "Add Milk" detectors each consist of a single sub-goal model (for "Pour Water" or "Add milk") and a "toying" model which corresponds to picking up the kettle or milk jug but not pouring water or milk into the mug. These detectors exploit the accelerometer and FSR outputs of the CICs attached to the kettle or milk jug, to indicate that this object has been picked up, moved, tilted, moved and put down, and the synchronised FSRs in the CIC attached to the mug to detect that at the time that the first object is tilted the mug begins to get heavier.
- The "Boil Water" detector has a single sub-goal HMM for "Boil Water" and a "toying" model. The sub-goal model uses the magnitude of the outputs from the accelerometer attached to the kettle body to pick up the movements caused by boiling water in the kettle.
- The "Fill Kettle" detector has a single sub-goal HMM for "Fill Kettle" and a "toying" model. The inputs to this detector are Gaussian neighbourhood values associated with the jug and kettle and the outputs of the CIC under the kettle to detect movement and an increase in weight.
- The "Stir" detector has a single sub-goal HMM for "Stir" and a "toying" model. The inputs to this detector are Gaussian neighbourhood values associated with the mug and the outputs of the CIC under the mug to detect movement.

The real-time CogWatch AR uses HMM file formats from the hidden Markov model toolkit (HTK) [25]. Thus HMM parameters can be optimised off-line using HTK and then transferred to the CogWatch AR. Figure 3 shows a screen-shot from the real-time action recognition system.

5) Scalability: Computational load is approximately proportional to $N_D \times N_F$, where N_D is the number of detectors, and N_F is the average number of features per detector. As the task becomes more complex the number of objects (and hence sensors) and sub-goals will increase, but the number of features per sub-goal is likely to remain approximately constant. In this case computational load will scale linearly with the number of sub-goals. In the limit, the computational techniques that enable real-time ASR for vocabularies on tens of thousands of words are applicable to this system. Hence computational load is unlikely to be an immediate issue.

VII. Experiments

A. Data Collection

Recordings were made at three different sites. A total of 38 participants, aged between 18 and 80, completed multiple individual sub-goals and full tea-making trials. In all cases synchronised CIC and Kinect outputs were recorded. In the full trial recordings, subjects were asked to make 4 different types of tea (as described in section II), as they would normally make it for themselves. These recordings were subsequently manually labelled using the data from the Kinect camera as guidance. In total, there are 1,124 recordings of isolated actions (4.01 hours) and 70 recordings of complete tea-making sessions (1.6 hours) (table I).

TABLE I. *Data used in AR development. Durations are in hours.*

Sub-goal	Trials	Dur.	Sub-goal	Trials	Dur.
Pour kettle	148	0.50	Stir	138	0.56
Add milk	69	0.22	Toy with kettle	26	0.07
Add sugar	220	0.40	Boil water	125	0.22
Add teabag	237	0.44	Toy with milk	30	0.11
Fill kettle	180	0.73			
Remove teabag	168	0.41	Full trial	70	1.6

B. Experimental procedure

Two experiments were conducted, namely recognition of isolated sub-goals and detection of sub-goals in full-trials. The isolated sub-goal recognition experiments used five-fold cross-validation, in which 20%, 20% and 60% of the recordings were used for testing, development and training, respectively. In the full trial experiments, all isolated sub-goal recordings were used for model training. The number of states in the sub-goal HMMs N ($5 \leq N \leq 60$) and the number of GMM components in the single-state "toying" model M ($1 \leq M \leq 512$), were determined empirically on the development data. Each state of the sub-goal HMM was associated with a single component Gaussian probability density function (PDF). Best results were achieved by using $N = 20, 20, 50$ and 70 states for the sub-goal model, and $M = 256, 512, 512$ and 32 GMM components for the "toying" models for "Front Actions" ("Add sugar", "Add tea-bag" and "Remove tea-bag"), 'Add Milk', 'Pour Kettle' and 'Fill Kettle' detectors, respectively. Figure 4 shows the effects of the number of sub-goal HMM states N and GMM components M in the "toying" model on recognition accuracy for the sub-goal "Add milk".

C. HMM parameter estimation

HMM parameter estimation was done using the HTK toolkit [25]. With the exception of the "Pour Kettle" and "Add Milk" sub-goals, the start and end times of sub-goals were identified manually. For "Pour Kettle" and "Add Milk" the

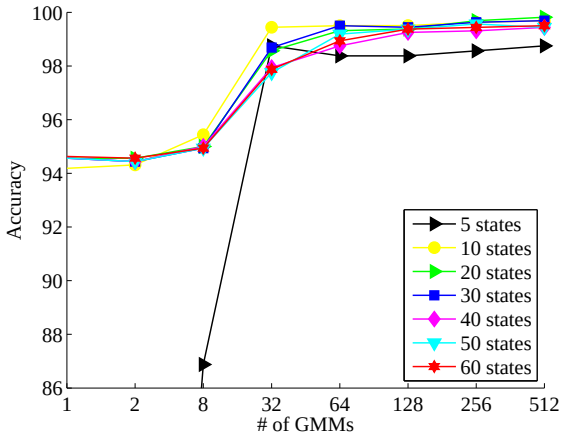

Fig. 4. *effects of the number of sub-goal HMM states N and GMM components M in the "toying" model on recognition accuracy for the sub-goal "Add milk".*

outputs of the FSR outputs were used to define the start and finish times as the moments of picking up and putting down the object. For initialization of an N state sub-goal HMM, each of the training recordings of sensor data for that sub-goal was divided into N equal segments, and the data in the n^{th} segments was used to estimate the mean and diagonal covariance matrix of the n^{th} HMM state. For the "toying" model, all of the recordings of the non-target sub-goals in the training set were used to estimate the mean and (diagonal) covariance matrix of a single Gaussian PDF.

The parameters of the sub-goal HMMs were optimised using the Baum-Welch algorithm [25]. The single Gaussian PDF associated with the "toying" HMM state was repeatedly divided and optimised using the E-M algorithm [25].

VIII. Results

The results of the recognition experiments on isolated sub-goals and full trials are shown in tables II and III, respectively. Detection accuracy for full trials is calculated as follows: A sub-goal occurring in a full trial is considered to have been correctly detected if and only if the sub-goal is detected by the corresponding detector and the detected and actual sub-goals overlap by 75%. If an actual sub-goal does not overlap with a detected sub-goal by 75% then a deletion (False Rejection (FR)) has occurred. If a detected sub-goal does not overlap with an actual sub-goal by 75%, then an insertion (False Alarm (FA)) has occurred. The % accuracy is given by:

$$\%Acc = \frac{Samples - Deletions - Insertions}{Samples} \times 100 \quad (3)$$

TABLE II. *Results of isolated sub-goal recognition experiments.*

Sub-goal	%Errors	Sub-goal	%Errors
Pour kettle	0.32	Add milk	0.21
Add sugar	0.09	Remove tea-bag	0.09
Boil water	1.65	Add teabag	0.09
Fill kettle	0.28		

TABLE III. *Results of full-trial sub-goal detection experiments (Ins = number of insertions, %Acc. = % Recognition accuracy, %FA = % Flase alarms, and %FR = % False rejections).*

Sub-goal	Samples	Correct	Ins	%Acc.	%FA	%FR
Pour kettle	53	53	0	100	0	0
Add milk	38	37	1	94.7	2.6	2.6
Add sugar	56	53	3	89.2	5.4	5.4
Remove tea-bag	60	56	6	83.3	10	6.7
Add tea-bag	60	58	5	88.3	8.3	3.3
Fill kettle	66	59	10	74.2	15.2	10.6
Stir	71	62	24	70	34	13

IX. Discussion

A. Recognition of isolated sub-goals

Table II shows that recognition for isolated sub-goals is very accurate, with an average error rate of just 0.39%. In other words, if the sub-goal boundaries are known then sub-goal recognition using the available sensor data is not difficult.

B. Sub-goal detection in full trials

Comparing tables II and III it is evident that the absence of information about sub-goal start and end times makes action recognition much more challenging. However, this is the relevant problem in real applications. The best performance is achieved for the sub-goals "Add Milk" and "Pour Kettle". These are the only sub-goals for which all of the objects that are involved are fully instrumented (i.e. fitted with a CIC).

Recognition of the sub-goals "Add Tea-bag", "Add Sugar" and "Remove Tea-bag" relies mainly on hand coordinate data from Kinect, plus small perturbations of the outputs from the CIC sensors attached to the mug caused by the weight-changes or movement due to adding a sugar cube or tea-bag to the mug, or removing a tea-bag from the mug. Given the minimal instrumentation performance is good.

The poorest performance is for "Stir". However, 50% of the false alarms occur at beginnings of instances of "Remove Tea-bag". Since "Remove Tea-bag" involves putting the spoon into the mug and moving it to pick up the tea-bag, the outputs of the mug CIC and the Kinect hand coordinates will be very similar to those for "Stir". Hence the insertion of "Stir" is to be expected. A solution would be to break down the sub-goals into smaller actions, so that "Stir" and the start of "Remove Tea-Bag" are both characterised by the same model.

The performance for "Fill Kettle" is also poor, with false alarm and false rejection rates of 15% and 11%, respectively. However, it is likely that these would be significantly improved if the water container were fitted with a CIC. In the current experiments, "Fill Kettle" relies on hand-location data and the outputs of the CIC attached to the kettle.

C. Object Centred Action Recognition

The results of these experiments point to an approach to action recognition based entirely on instrumented objects. From this perspective, actions are characterised in terms of how the objects involved "experience" them through their sensors. Hence it is appropriate to refer to this as an "object-centred" approach to action recognition, to differentiate it from an "environment-oriented" approach in which, for example, action recognition is achieved by the application of image processing to the scene.

Object-centred action recognition has a number of practical advantages. It is very easy to deploy in a real environment, such as a kitchen, because all that is required is the introduction of instrumented tools and objects into the environment. In addition, no explicit calibration of the sensors is required. Any calibration that is required to compensate for drift in the sensor outputs can be done while the object is at rest, without involving the user.

D. Application to rehabilitation for AADS

The AR system described in this paper has been incorporated into the CogWatch system for cognitive rehabilitation of stroke patients with AADS [11], and future publications will describe its performance on patient data. The performance of the AR system described in this paper will improve as a result of further research. However, some errors are inevitable. In the CogWatch system the output of the AR system is passed to a Task Model, based on a Markov Decision Process (MDP) [19] or Partially Observable MDP (POMDP) [15]. In the case of a POMDP, the task model includes a statistical characterisation of the errors that the AR makes and is therefore able to accommodate them to some degree.

X. Conclusion

This paper presents a novel HMM-based architecture for AR that can accommodate the lattice of sub-goals that describes an ADL such as tea-making. The results show that HMMs combined with instrumented objects provide a viable approach to action recognition. Best performance is obtained when all of the objects are instrumented. This suggests an object-centred approach to action recognition, in which an action is characterised in terms of the ways that the objects involved "experience" it through their sensors. Object-centric action recognition is particularly attractive for practical applications because of its ease of deployment. Future publications will report the results obtained by the system on patient data.

Acknowledgment

The authors wish to thank all CogWatch partners for their contributions. This work has been supported by the European Commission under the grant FP7-ICT-2011-288912.

References

[1] Amft, O. and Tröster, G.,. "Recognition of dietary activity events using on-body sensors", Artificial Intelligence in Medicine, 42, 121-136, 2008.

[2] Annett, J., K. D.Duncan, R. B.Stammers, and M.Gray, "Task Analysis", London: HMSO, 1971.

[3] Barbieri, E., and Renzi, E. D., "The executive and ideational components of apraxia". Cortex , pp. 535544., 1988.

[4] Berlin, E., Liu, J., van Laerhoven, K. and Schiele, B., "Coming to Grips with the Objects We Grasp: Detecting Interactions with Efficient Wrist-Worn Sensors", Proc. TEI 2010, 57-64, 2010.

[5] Bickerton, W., Riddoch, M., Samson, D., Balani, A., Mistry, B., and Humphreys, G., "Systematic assessment of apraxia and functional predictions from the Birmingham cognitive screen". Journal of Neurology, Neurosurgery, and Psychiatry, pp. 513521, 2012.

[6] Cogollor, J.M., Hughes, C., Ferre,M., Rojo, J., Hermsdörfer, J., Wing, A. and Campo, S.,"Handmade Task Tracking Applied to Cognitive Rehabilitation", Sensors, 12(10): 1421414231, 2012.

[7] Gales, M. and Young, S.J., "The Application of Hidden Markov Models in Speech Recognition", Foundations and Trends in Signal Processing, Vol. 1, No. 3, 195304, 2007.

[8] Gellersen, H-W., Beigl, M. and Krull, H., "The MediaCup: awareness technology embedded in an everyday object", In H-W. Gellersen (ed.) Handheld and Ubiquitous Computing 1st International Symposium HUC99, Berlin: Springer, 308-310, 1999.

[9] Genest, A., Gutwin, C., Tang, A., Kalyn, M., and Ivkovic, Z., "KinectArms: a Toolkit for Capturing and Displaying Arm Embodiments in Distributed Tabletop Groupware", Proc. Conference on Computer Supported cooperative work (CSCW'13), 157-166, 2013.

[10] Hasan, M. K., Rubaiyeat, H. A., Lee, Y. K., and Lee, S., "A reconfigurable HMM for activity recognition". In Advanced Communication Technology 20 ICACT 2008. 10th IEEE Int. Conf on , Vol. 1, pp. 843-846, 2008.

[11] Hermsdörfer, J., Bienkiewicz, M., Cogollor, J. M., Russell, M., Jean-Baptiste, E., Parekh, M., Wing, A. M., Ferre, M., Hugues, C., "CogWatchAutomated Assistance and Rehabiliation of Stroke-induced Action Disorders in the Home Environment", HCI Aspects of Optimal Healing Enviroments, 2013.

[12] Hondori, H.M., Khademi, M. and Lopes, C.V., "Monitoring intake gestures using sensor fusion (Microsoft Kinect and interial sensors) for smart home tele-rehab setting", Proceedings of the 1st Annual Healthcare Innovation Conference, 2012.

[13] Hong, Y.-J., Kim, I.-J., Ahn, S. C., and Kim, H.-G., "Activity Recognition Using Wearable Sensors for Elder Care". Second IEEE Int. Conf. on Future Generation Communication and Networking, 302305, 2008.

[14] Hughes, C.M., Baber, C., Bienkiewicz, M. and Helmsdörfer, "Application of human error identification (HEI) techniques to cognitive rehabilitation in stroke patients with limb apraxia", in Access in Human-Computer Interaction. Applications and Services for Quality of Life, 463-471, Springer Berlin Heidelberg, 2013.

[15] Jean-Baptiste, E.M.D., Nabiei, R., Parekh, M., Fringi, E., Drozdowska, B., Baber C., Jančovič, P., Rotshein, P. and Russell, M., "Intelligent Assistive System Using Real-Time Action Recognition for Stroke Survivors", Proc. IEEE Int. Conf. Healthcare Informatics, Verona, 2014.

[16] Junker, H., Lukowicz, P. and Troster, G., "Continuous recognition of arm activities with body-worn inertial sensors", In Proc. 8th IEEE International Symposium on Wearable Computers, Vol. 1, 188-189, 2004.

[17] Liu, K., Chen, C., Jafari, R. and Kehtarnavaz, N., "Multi-HMM classification for hand gesture recognition using two differing modality sensors", Proc. 10th IEEE Dallas Circuits and Systems Conference (DCAS), 2014.

[18] Maekawa, T., and Watanabe, S., "Unsupervised Activity Recognition with User's Physical Characteristics Data", In Proc. 15th IEEE International Symposium on Wearable Computers , 89-96, 2011.

[19] Puterman, M., "Markov decision processes: discrete stochastic dynamic programming", Wiley, 2008.

[20] Rabiner, L.R., "A tutorial on hidden Markov models and selected applications in speech recognition", Proc. IEEE, vol. 77, no. 2, 1989.

[21] Spohrer, J.C., Brown, P.F., Hochschild, P.H. and Baker, J.K., "Partial traceback in continuous speech recognition ", Proc. IEEE Int. Cong. Cybernetics and Society, 1980.

[22] Stiefmeier, T., Roggen, D., Tröster, G., Ogris, G. and Lukowicz, P., "Wearable Activity Tracking in Car Manufacturing", Pervasive Computing, 42-50, April-June 2008.

[23] Wagner, J., Ploetz, T., Halteren, A. V., Hoonhout, J., Moynihan, P., Jackson, D., Ladha, C., "Towards a Pervasive Kitchen Infrastructure for Measuring Cooking Competence". Proc Int. Conf Pervasive Computing Technologies for Healthcare, 107-114, 2011.

[24] Wu, J., Osuntogun, A., Choudhury, T., Philipose, M. and Rehg, J.M., "A Scalable Approach to Activity Recognition based on Object Use", Proc. 11th IEEE Int. Conf. Computer Vision, 2007.

[25] Young, S., Evermann, G., Kershaw, D., Moore, G., Odell, J., Ollason, D., Povey, D., Valtchev, V. and Woodland, P., "The HTK Book (version 3.2)", Cambridge University Engineering Department, 2002.

[26] The Stroke Association website, *http://www.stroke.org.uk/about-stroke*.

[27] CogWatch: Cognitive rehabilitation of apraxia and action disorganisation syndrome, *http://www.cogwatch.eu/*.