CogWatch – Cognitive Rehabilitation of Apraxia and Action Disorganization Syndrome

D3.2.2 Report on data analysis for action recognition II

Deliverable No.		D3.2.2	
Workpackage No.	**WP3**	Workpackage Title	**Activity Recognition & Prediction**
Task No.	**T3.2**	Activity Title	**Action Recognition**
Authors (per company, if more than one company provide it together)		**Author (s) Name (s) (Company)** **Martin Russell, Roozbeh Nabiei, Manish Parekh, Thomas McKay-Smith, Chris Baber (UOB)** **Philipp Gulde, Joachim Hermsdörfer (TUM)**	
Status (F: final; D: draft; RD: revised draft):		**F**	
File Name:		**Cogwatch_D3.2.2_DataAnalysis_AR_II**	
Project start date and duration		**01 November 2011, 40 Months**	

EXECUTIVE SUMMARY

The main objective of the work package WP3 “Activity Recognition & Prediction” is to explore psychological and pattern-based models to construct action recognition techniques, apply these techniques to the data collected from the patient studies to develop an action recognition system, review psychological models and apply advanced statistical methods to provide a reliable action prediction model that will be able to identify patients’ intentions, predict actions and assess the progress of a task. The activities in this work package are divided into three main tasks (1) description and selection of action recognition techniques, (2) data collection methods, (3) action prediction models

Many of these objectives and tasks have been addressed in the previous deliverables D3.1, D3.2.1 and D3.3.1. Current developments in action prediction models are addressed in the deliverable D3.3.2 which is compiled simultaneously. The present deliverable addresses the further development of action recognition techniques and the empirical analysis of related data.

Action recognition basing on hidden Markov models (HMMs) has been further developed. A feature vector containing all relevant signals from the coasters and the Kinect presents the input into the models. It was decided to realize the action recognition (AR) at the hierarchical level of sub-goals. Consequently ten different sub-goals have to be recognized for the tea making task. A novel, parallel Viterbi decoder with a partial trace-back algorithm was implemented to enable real-time behavior.

AR was tested empirically using data of 26 healthy subjects who have completed the sub-goals or the complete trial of the tea making task several times. Evaluating the separated sub-goals, those sub-goals were detected with a particularly high accuracy (> 95%), for which the coaster data, signaling object movement, provided all relevant task information. The accuracy decreased for sub-goals in which coaster data were only partial informative. Error rates were highest (approximately 20%) for sub-goals which only involved hand trajectory data. Continuing development will further decrease latencies and improve detection also within complete sequences.

The kinematics of both hands during the execution of the tea making task was analyzed in 7 stroke patients and in 9 healthy control subjects. The comparison of unimanual with bimanual task performance yielded interesting clinical findings in patients, for example: Does the right hand in patients with left brain damage lead performance during bimanual execution despite the presence of paresis? Concerning AR, characteristic profiles were found for the individual segments emphasizing a possible use of kinematics for online analysis. Patients deviated in durations and velocities but not in spatial variables from controls. In addition, deficits were similar across the different segments. This indicates that a generalization of findings and rules in healthy subject on patients’ behavior is possible with some restrictions.

For the second CogWatch task “tooth brushing” a number of sensors measuring position or acceleration can be considered for indicating the position of the brush in the mouth. A particularly elegant method may be the analysis of the sound of the brush measured with a microphone outside the mouth. This idea was investigated using Gaussian Mixture Models basing on the frequency information. With multiple data from one individual it was shown that various relevant dimensions like front – back, top – bottom, left – right, and inside – outside could be distinguished with high precision above 85% up to nearly 100%. The

precision under less ideal conditions still has to be evaluated. The technique may also be combined with alternate approaches to reach high detection accuracies.

TABLE OF CONTENTS

Contents

1. INTRODUCTION 12

1.1 Activity recognition 12

1.2 Objectives since month 11 12

2. REAL-TIME ACTION RECOGNITION IN TEA-MAKING 14

2.1 The tea-making task 14

2.1.1 Terminology 14

2.1.2 Implications for activity recognition 15

2.2 The first prototype CogWatch Action Recognition system 15

2.2.1 Outline of the system 15

2.2.2 Instrumentation 16

2.2.3 Sensor data capture and pre-processing 17

2.2.4 Structure of the AR system 18

2.2.5 Structure of a sub-goal detector 19

2.2.6 Recognition strategy for the sub-goals of tea-making 20

2.2.7 Real-time recognition 21

2.2.8 Partial Trace-back 21

2.2.9 Latency 21

2.2.10 Implementation of the real-time AR 22

2.2.11 Validation 23

2.3 Off-line experimental evaluation 23

2.3.1 Data used in the evaluation 23

2.3.2 Effect of combining sensor-based and Kinect-based features 24

2.3.3 Recognition of “add tea-bag”, “add sugar” and “remove tea-bag” 25

2.3.4 Effect of 50Hz sample rate 27

2.4 Summary of real-time Action Recognition for tea-making 27

2.4.1 AR System 27

2.4.2 AR Performance 28

2.4.3 Real-time issues 28
2.4.4 Future Work 28
2.5 Exploitation 29
3. SEGMENTATION AND KINEMATIC-BASED RECOGNITION OF TEA MAKING IN HEALTHY SUBJECTS UND PATIENTS WITH ADL DEFICITS 30
3.1 Introduction 30
3.2 Materials and Methods 31
3.2.1 Task and Procedure 31
3.2.2 Segmentation 32
3.2.3 Kinematic analysis 35
3.3 Results 35
3.4 Discussion 42
4. ACTION RECOGNITION FOR TOOTH BRUSHING 46
4.1 Acoustic identification of tooth-brush position location 46
4.1.1 The second CogWatch prototype system 46
4.1.2 Categorization of mouth positions 46
4.1.3 Automatic identification of toothbrush head location 46
4.2 A corpus of recordings of tooth-brushing 47
4.2.1 Audio data collection 47
4.2.2 Example acoustic signals for tooth brushing 48
4.2.3 Initial conclusions 50
4.3 Automatic identification of tooth-brushing position from audio data 50
4.3.1 Method 50
4.3.1.1 Feature extraction 50
4.3.1.2 Statistical modelling 51
4.3.2 Experiments 52
4.3.2.1 Experiment 1: Two-classes, front vs back 52
4.3.2.2 Experiment 2: Two-classes, top vs bottom 53
4.3.2.3 Experiment 3: Three classes, back-right, back-left and front 54
4.3.2.4 Experiment 4: Four classes, back-right, back-left, front-inside and front-outside. 55

4.3.2.5 Experiment 5: Six classes – front-inside, front-outside, bottom-back-left, bottom-back-right, top-back-left and top-back-right. 56

4.3.2.6 Experiment 6: Eight classes – bottom-front-inside, bottom-front-outside, top-front-inside, top-front-outside, bottom-back-left, bottom-back-right, top-back-left and top-back-right. 57

4.3.3 Summary of classification experiment results 58

5. REFERENCES 60

TABLE OF FIGURES

Figure 1: Hierarchical tree representation of the tea making task (from D1.1 Report on scenarios) 14

Figure 2: Outline of the CogWatch Action Recognition system 16

Figure 3: Guassian neighborhoods to detect the proximity of the user's hand to an object in the tea-making task. 17

Figure 4: Feature synchronisation in the CogWatch Action Recognition system 18

Figure 5: Architecture of the CogWatch Action Recognition system 19

Figure 6: Structure of an individual sub-goal detector 20

Figure 7: Explanation of the Partial Traceback algorithm to enable real-time Viterbi decoding. 23

Figure 8: Comparison of recognition accuracy for the sub-goal "fill kettle" using just CIC sensor information (left) and CIC sensor information plus Kinect hand coordinates (right). 25

Figure 9: Recognition accuracy for the three "front actions" using Kinect hand coordinate data only. 26

Figure 10: Recognition accuracy for the three "front actions" using Kinect hand coordinate data plus the outputs of the FSRs in the mug CIC. 26

Figure 11: Recognition accuracy for the three "front actions" using Kinect hand coordinate data and derivatives, plus the outputs of the FSRs in the mug, milk jug and kettle CICs. ... 27

Figure 12**:** Experimental setting for the tea-making task including a water jug, milk, a plate for used teabags, teabags, sugar, coffee, a kettle, a mug and a spoon. 31

Figure 13: Example of the synchronization of data streams with action segmentation of the tea-making task (bimanual). 33

Figure 14: Hand velocity and segments identified for a patients' trial. Note that only segments 1, 2, 3, 4, 6 & 7 were executed and their order was mixed. 34

Figure 15: Number of sub-segments performed per trial and probabilities of occurrence of the different sub-segments in the control group and the patient groups (LBD, RBD). 34

Figure 16: Movement time in total for the different task conditions in controls. 35

Figure 17: Movement time for the different sub-segments in the bimanual task condition in controls. 35

Figure 18: Overall path length for the different task conditions in controls. 36

Figure 19: Path length for the different sub-segments in the bimanual task condition in controls. 36

Figure 20: Maximum velocity peak in total for the different task conditions in controls. 37

Figure 21: Maximum velocity peak for the different sub-segments in the bimanual task condition in controls 37

Figure 22: Movement time in total for the different task conditions in controls, patients with left brain damage and patients with right brain damage. 38

Figure 23: Movement time for the different sub-segments in the three task conditions in controls, LBD & RBD patients. 38

Figure 24: Path length in total for the different task conditions in controls, patients with left brain damage and patients with right brain damage. 39

Figure 25: Path lengths for the different sub-segments in the three task conditions in controls, LBD & RBD patients. 40

Figure 26: Maximum velocity peak in total for the different task conditions in controls, patients with left brain damage and patients with right brain damage. 41

Figure 27: Maximum velocity peaks for the different sub-segments in the three task conditions in controls, LBD & RBD patients. 41

Figure 28: Movement time displayed against path length in the bimanual task condition for the dominant / ipsilesional hand in controls, LBD & RBD patients including a linear regression line for each group. 42

Figure 29: Hierarchical description of the partition of the mouth cavity into different location for tooth-brushing. 46

Figure 30: Spectrogram of a five second recording of brushing teeth in the back-bottom-left region of the mouth. 47

Figure 31: Spectrogram of a five second recording of brushing teeth in the top-front-outside region of the mouth 49

Figure 32: Spectrogram of a five second recording of brushing teeth in the top-front-inside region of the mouth 50

Figure 33: Results of classification experiments to distinguish between tooth-brushing at the front and back of the mouth. 52

Figure 34: Results of classification experiments to distinguish between tooth-brushing at the top and bottom of the mouth. 53

Figure 35: Results of classification experiments to distinguish between tooth-brushing at the back-right, back-left and front of the mouth. 54

Figure 36: Results of classification experiments to distinguish between tooth-brushing at the back-right, back-left, front-inside and front-outside of the mouth. 55

Figure 37: Results of classification experiments to distinguish between tooth-brushing at six different locations: the bottom-back-right, bottom-back-left, top-back-right, top-back-left, front-inside and front-outside of the mouth. 56

Figure 38: Results of classification experiments to distinguish between tooth-brushing at eight different locations: the bottom-front-inside, bottom-front-outside, top-front-inside, top-front-outside, bottom-back-left, bottom-back-right, top-back-left and top-back-right. 57

LIST OF TABLES

Table 1: Details of recordings of data used in the experiments reported in this section. 24
Table 2: Demographic and clinical data of patients tested in the tea-making task. 32
Table 3: Statistics of the recordings of tooth-brushing that were made for the pilot experiment. 48
Table 4: Recordings used in experiment 1. 52
Table 5: Recordings used in experiment 2. 53
Table 6: Recordings used in experiment 3. 54
Table 7: Recordings used in experiment 4. 55
Table 8: Recordings used in experiment 5. 56
Table 9: Recordings used in experiment 6. 57
Table 10: Summary of classification results. 58

REVISION HISTORY

Revision no.	Date of Issue	Author(s)	Brief Description of Change
Version 1	24/07/2014	TUM	First draft for sending to UOB
Version 2	11/08/2014	UOB	UOB parts added
Version 3	28/08/2014	TUM	Parts added and format combined
Version 4	1/09/2014	TUM	Changes from UOB integrated and sent to internal quality management
Final	02/10/2014	TUM	Final version, suggestions from internal review integrated

LIST OF ABBREVIATIONS AND DEFINITIONS

Abbreviation	Abbreviation
AADS	**Apraxia and Action Disorganization Syndrome**
ADL	**Activity of daily living**
AR	**Activity Recognition**
ASR	**Automatic speech recognition**
CIC	**CogWatch Instrumented Coaster**
CSC	**Contention scheduling system**
DBN	**Dynamic Bayesian network**
FMEA	**Failure Modes Effects Analysis**
FSH	**Force Sensitive Handle**
FSR	**Force Sensitive Resistor**
GMM	**Gaussian mixture model**
HMM	**Hidden Markov model**
IAN	**Interactive action model**
MDP	**Markov decision process**
PDF	**Probability density function**
POMDP	**Partially observable Markov decision process**
PSR	**Pressure Sensitive Resistor**
RFID	**Radio Frequency Identification**
SAS	**Supervisory attentional system**
TM	**Task model**

1. INTRODUCTION

1.1 Activity recognition

The main objective of the work package WP3 “Activity Recognition & Prediction” is to explore psychological and pattern-based models to construct action recognition techniques, apply these techniques to the data collected from the patient studies to develop an action recognition system, review psychological models and apply advance statistical methods to provide a reliable action prediction model that will be able to identify patients’ intentions, predict actions and assess the progress of a task. The activities in this work package are divided into three main tasks (1) description and selection of action recognition techniques, (2) data collection methods, (3) action prediction models

Many of these objectives and tasks have been addressed in the previous deliverables D3.1, D3.2.1 and D3.3.1. The deliverable D3.3.2, which is produced in parallel with the present deliverable, addresses action prediction in continuation of D3.3.1. The present work addresses action recognition and in particular the analysis of data assessed within this context.

Activity Recognition (AR) in the CogWatch project refers to technology for monitoring of a participant engaged in an activity for everyday living (ADL). The CogWatch AR system should recognize the action that the participant is performing. Together with the CogWatch action prediction system it should know the state of the action. It should be able to estimate the likelihood of successful completion, and it should be able to synthesize useful cues and feedback to the participant to redirect action. The CogWatch AR system monitors an activity using sensors attached to tools and objects, plus video-based estimates of the participant’s hand positions.

The deliverable addresses the three main tasks of related empirical work namely “Video-based action recognition (T3.2.1)”, “Marker-based action recognition (T3.2.2)”, and “Object-based action recognition (T3.2.3)”.

Video and depth sensor information of the Kinect is used to calculate the positions of the hand that provides an important source of information for action recognition. Approaches based on the computer vision were dropped due to accuracy limitations (see Progress Report 2).

Object-based action recognition turned out as a reliable and suitable methodology which is now extensively used in the Cogwatch prototypes.

Markers-based action recognition refers to motion recording of body segments, mainly the hand, and the corresponding kinematical analyses.

1.2 Objectives since month 11

Three sets of objectives are described in the rest of this report. Section 2 describes the further development of an action recognition (AR) approach based on Hidden Markov Models for the “Tea making” CogWatch prototype (Section 2). The feature vector containing all information for the instrumented coasters as well as hand trajectory information from Kinect has been defined. Within the hierarchical actions levels, the sub-goal level has been

suggested to be most promising for AR, but AR on lower levels is possible as well. The need for real time encoding as well as the fact that the sequence of sub-goals is not fixed and, in the case of bimanual operation, sub-goals my additionally occur in overlapping time during daily activities like tea making has put particular demands on the development. Specific algorithms and procedures were developed for that purpose and implemented. Empirical tests of the implemented AR were conducted using data of healthy subjects.

An evaluation of marker based motion recording is presented in Section 3. AR based on segmentation of the multi-step action and kinematic analyses were tested in healthy subjects and a group of AADS patients. The investigation served the purpose to determine the degree the results obtained in healthy subject about the quality of AR can be generalized to AADS patients. In addition, the potential support of AR by kinematic measures was evaluated. The feasibility of this approach that is currently off-line but may be on-line in future version of the CogWatch system is discussed.

Section 4 describes the evaluation of an approach to AR for the second CogWatch task “Tooth brushing”. Specifically, the acoustic sound of the tooth-brush in the mouth was measured and algorithms were developed to infer the position of the brush in the mouth. Promising results were obtained from this approach.

2. REAL-TIME ACTION RECOGNITION IN TEA-MAKING

2.1 The tea-making task

The tea-making task and the rationale for choosing it as the task for the first CogWatch prototype system are described fully in CogWatch deliverable D1.1 “Report on scenarios”. Figure 1 (taken from D1.1) shows a hierarchical tree based description of the task. It is included here for ease of reference.

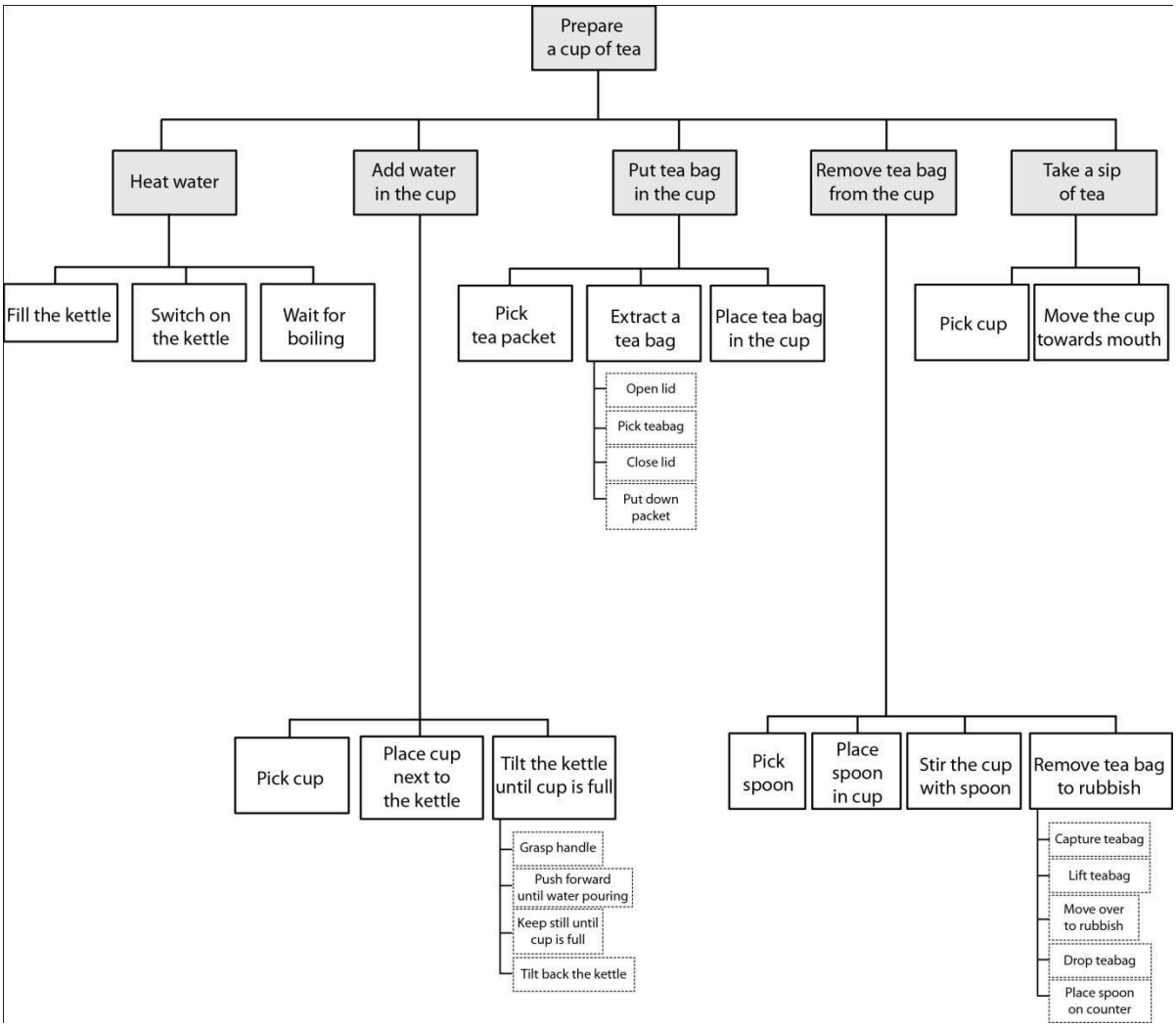


Figure 1: Hierarchical tree representation of the tea making task (from D1.1 Report on scenarios)

2.1.1 Terminology

The following terminology from D1.1 is used to describe items in this hierarchical tree:

- The goal corresponds to the root of the tree and is “prepare a cup of tea”. The whole description is aimed at achieving the goal (which, therefore, can be defined in terms of completion)
- The items at the next level of the tree, namely “heat water”, “add water to cup”, “put tea bag in the cup”, “remove the tea bag from the cup” and “take a sip of tea”, will be referred to as sub-goals.
- The items at the third level of the tree, for example “fill the kettle”, “switch on the kettle” and “wait for boiling”, will be referred to as tasks.
- The leaves of the tree, for example “grasp handle”, “push forward until water pouring”, “keep still until cup is full” and “tilt back the kettle”, will be referred to as sub-tasks.

2.1.2 Implications for activity recognition

The choice of tea-making as the scenario for the first CogWatch prototype system has a number of implications. First, the description of tea-making as a hierarchical tree facilitates the use of methods from other fields, such as automatic speech recognition (ASR). However, the sequential structure of the tea making activity, and in particular whether its components can be thought of as a well-ordered or partially-ordered set, has implications for the choice of architecture for the action recognition system. These issues are discussed in more detail below, in the description of the approach to action recognition that was taken.

2.2 The first prototype CogWatch Action Recognition system

2.2.1 Outline of the system

Figure 2 is a diagram of the CogWatch Action Recognition (AR) system for the tea-making task. AR is performed by analyzing the outputs of sensors attached to the principal objects involved in the task, namely the mug, milk-jug and kettle. These sensors indicate whether an object is moving or stationary, whether it has been tilted, whether it is resting on the work-surface or held above the surface, and the object’s weight when resting on a surface. The user interacts with the instrumented objects and the signals from these plus hand coordinates, obtained from a Microsoft Kinect system, are passed to the real-time AR system. When the user completes a sub-goal, it is recognized and its identity is passed to the Task Model.

The function of the Task Model is to track the user’s progress through the task, and to detect an error if it occurs. The Task Model in the first CogWatch prototype is based on a Markov Decision Process (MDP). Every sequence of sub-goals that can be continued to a satisfactory completion of the task is a state of the Task Model. Associated with each state is an optimal strategy, which is the best sub-goal for the user to execute next. The optimal strategy is pre-computed before the system is exposed to a user.

If an error has been detected the Task Model passes an error code to the Error table / cueing module, which determines whether or not a cue is to be passed to the user, and if so the type of cue. Even if an error has not occurred the Task Model knows the best next action for the user (the „optimal strategy“). This can be passed to the Error-table / cueing module which may or may not pass it to the user (recall that CogWatch is a rehabilitation rather than an assistive system). The Task Model is described in detail in deliverable D3.3.2.

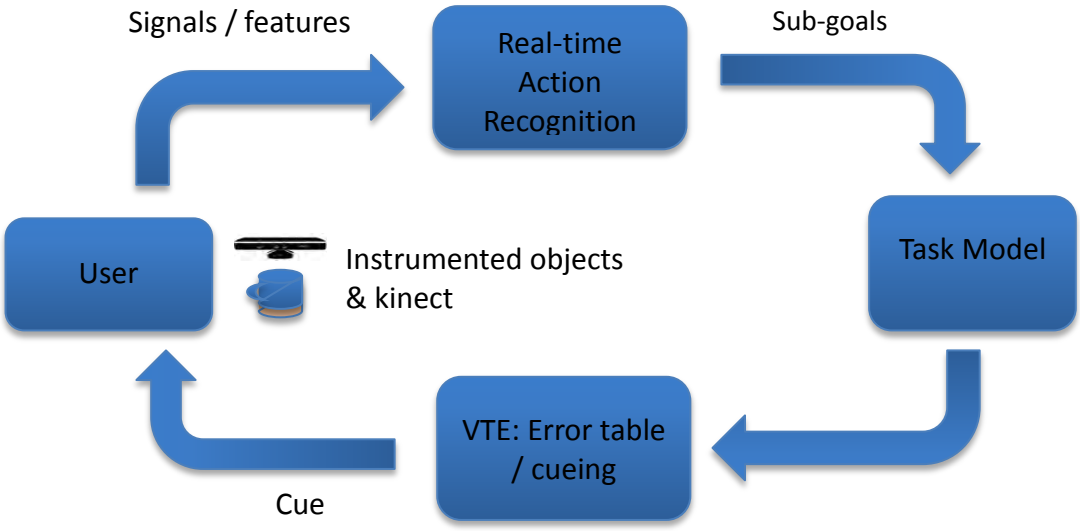


Figure 2: Outline of the CogWatch Action Recognition system

2.2.2 Instrumentation

The approach to AR used in the CogWatch prototype relies primarily on interpreting the outputs of sensors attached to the objects involved in the task. This differentiates it from other approaches that rely, for example, on advanced image processing. In the tea-making task the sensors are encased in a 'coaster' which fits underneath a mug or jug. This is the CogWatch Instrumented Coaster (CIC) which is described fully in D2.2.2.

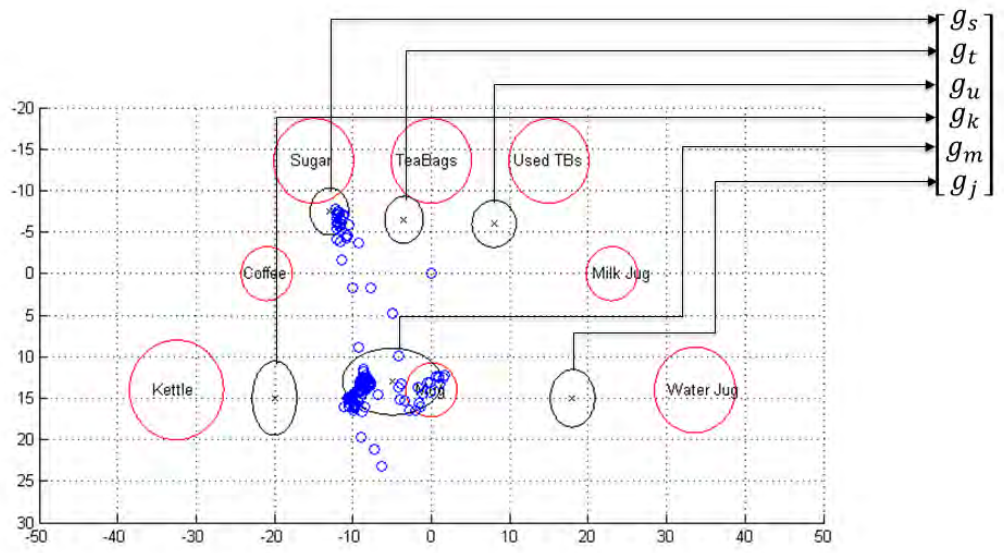


Figure 3: Gaussian neighborhoods to detect the proximity of the user's hand to an object in the tea-making task.

Briefly, the CIC contains a three-axis accelerometer (to measure the object's movement), three force-sensitive resistors (FSRs) (to detect whether the object is resting on a surface and to estimate weight), a battery and a Bluetooth module. The CIC sensor outputs are updated 50 times per second.

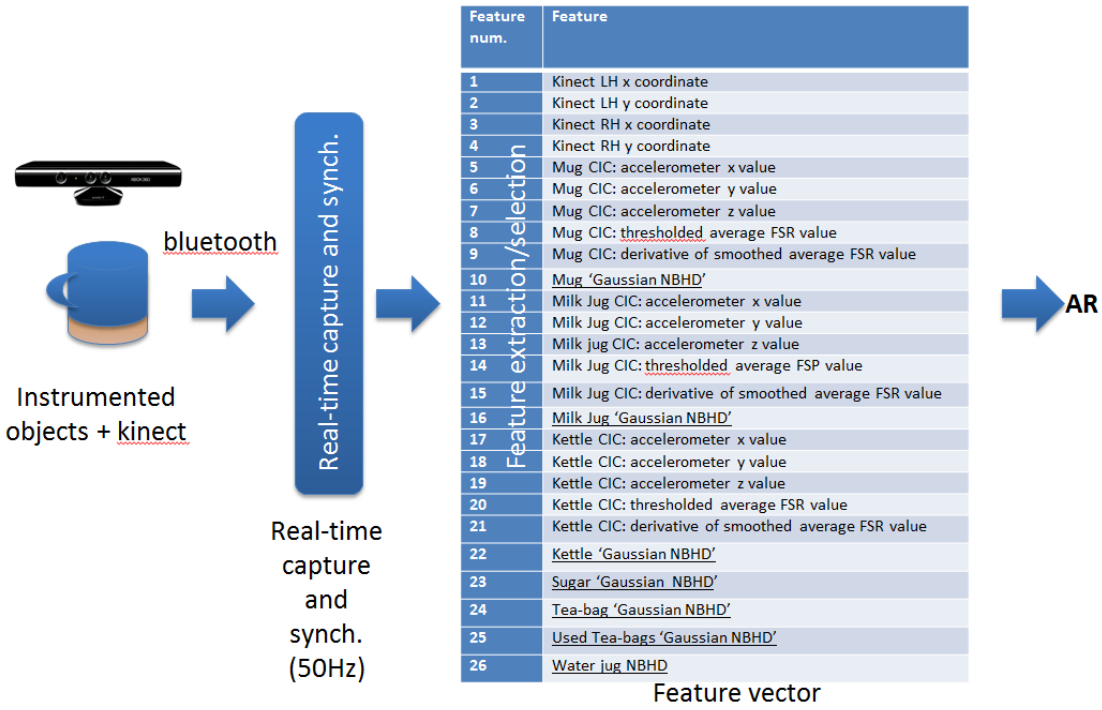
In addition to the CICs, the AR uses hand position coordinates, measured using the Microsoft Kinect system, to characterize the user's interaction with objects that are not fitted with a CIC (namely the tea-bag, sugar and user-tea-bag containers and the water jug). The hand coordinates are used only to detect proximity of the user's hand to an object, and the details of the hand's trajectory during the task are ignored. The method is based on the notion of "Gaussian neighborhoods". For each object the distribution of hand coordinates when the user interacts with that object is learnt and modelled as a Gaussian probability density function (PDF). For each object the Gaussian neighborhood gives rise to a feature, whose value is zero when the user's hand is not interacting with the object and increases as it approaches the object. The Gaussian neighborhoods in the tea-making task are shown in Figure 3.

For example, in Figure , as the user's hand approaches the sugar container the value of the feature g_s increases from 0 to 1.

2.2.3 Sensor data capture and pre-processing

Figure shows the capture and synchronization of sensor data that is the first stage in the AR process.

The left hand side of the figure represents the sensors involved, namely the CICs attached to the objects involved in the task, and Kinect. The outputs of these sensors are transmitted via a wireless Bluetooth channel, captured in real-time, synchronized and compiled into a feature vector similar to that shown in the figure. The feature vector is updated and passed to the real-time AR module 50 times per second.



Feature num.	Feature
1	Kinect LH x coordinate
2	Kinect LH y coordinate
3	Kinect RH x coordinate
4	Kinect RH y coordinate
5	Mug CIC: accelerometer x value
6	Mug CIC: accelerometer y value
7	Mug CIC: accelerometer z value
8	Mug CIC: thresholded average FSR value
9	Mug CIC: derivative of smoothed average FSR value
10	Mug 'Gaussian NBHD'
11	Milk Jug CIC: accelerometer x value
12	Milk Jug CIC: accelerometer y value
13	Milk jug CIC: accelerometer z value
14	Milk Jug CIC: thresholded average FSP value
15	Milk Jug CIC: derivative of smoothed average FSR value
16	Milk Jug 'Gaussian NBHD'
17	Kettle CIC: accelerometer x value
18	Kettle CIC: accelerometer y value
19	Kettle CIC: accelerometer z value
20	Kettle CIC: thresholded average FSR value
21	Kettle CIC: derivative of smoothed average FSR value
22	Kettle 'Gaussian NBHD'
23	Sugar 'Gaussian NBHD'
24	Tea-bag 'Gaussian NBHD'
25	Used Tea-bags 'Gaussian NBHD'
26	Water jug NBHD

Figure 4: Feature synchronization in the CogWatch Action Recognition system

The values in the feature vector include raw processed versions of the sensor outputs. For example, whether the object is resting on a surface or suspended above the surface can be detected by applying a simple threshold to the average value of the FSRs, whereas more subtle weight changes (for example, resulting from pouring water or milk into a mug, or from removing a used tea-bag) are indicated by the derivative of the average FSR value.

2.2.4 Structure of the AR system

The architecture of the AR system is shown in Figure 5. The AR consists of a parallel set of detectors, with dedicated detectors for each of the sub-goals of the tea-making task and for important actions that the user might perform that are not sub-goals, such as "toying" with a kettle of boiling water.

The sub-goal detectors continuously monitor the input feature vectors in parallel in real time. When a detector judges that its sub-goal has occurred the AR outputs a label for that sub-goal, which is passed to the Task Model. The parallel architecture was chosen to cope with a situation where the user executes two sub-goals at the same time, or in overlapping time. A more conventional decoder that assumes that events occur in a well-ordered sequence would be unable to accommodate this type of behavior.

Each sub-goal detector uses only the relevant features from the feature vector. For example, the detector for the sub-goal "add milk to mug" uses the features from the CICs attached to the milk jug and the mug, plus the corresponding Gaussian neighborhood values.

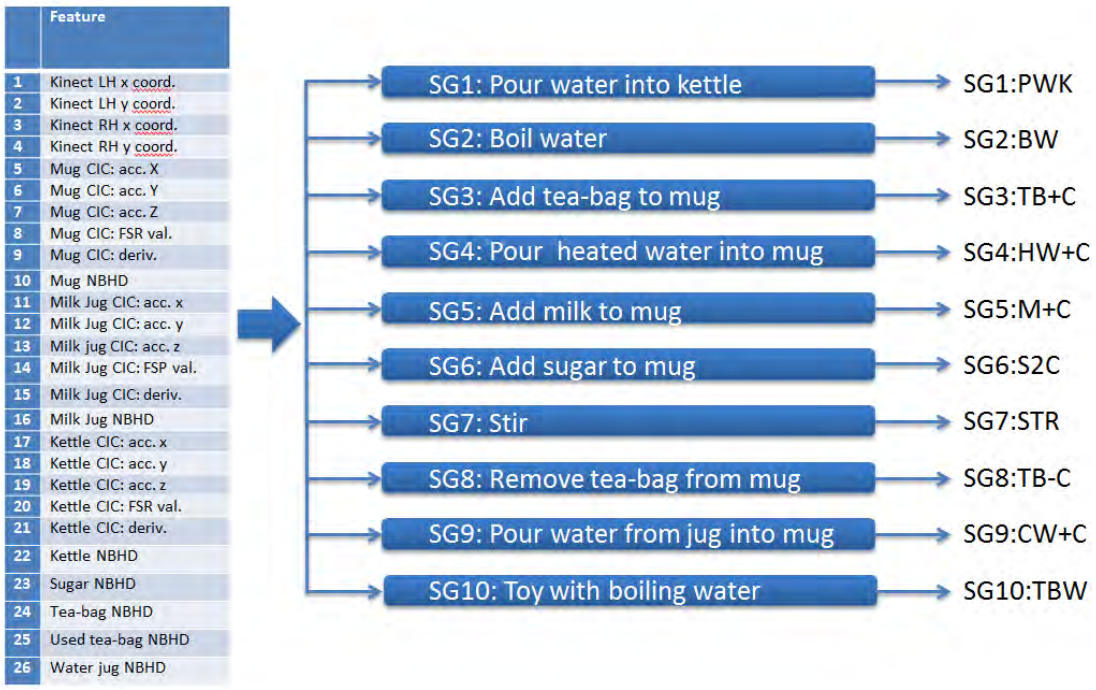


Figure 5: Architecture of the CogWatch Action Recognition system

2.2.5 Structure of a sub-goal detector

Figure 6 represents an individual sub-goal detector for the sub-goal “pour milk” (from the milk jug into the mug). A “feature map” for this detector indicates which of the features in the feature vector are relevant to this sub-goal. The detector is based on two hidden Markov models (HMMs), namely a sub-goal model and a toying model. HMMs can be thought of as generic statistical models for sequential data. They are most commonly used in automatic speech recognition but are equally suited to statistical modelling of other types of sequential data. The rationale for using HMMs for sub-goal recognition in the CogWatch AR system was explained in a previous report (D3.1.1).

The sub-goal HMM is a statistical model of the variations of sequences of sensor outputs that occur when a user executes that sub-goal. It consists of a sequence of states, which can be thought of as models of the sequence of tasks and sub-tasks that make up the sub-goal. Each of these states is associated with a Gaussian Mixture Model (GMM) that characterizes the distribution of sensor outputs for that task or sub-task. The “toying” model is a single GMM, with a large number of components, that models the values of the sensor outputs that occur when the user is not executing the sub-goal. These two models are run in continuous competition. An output occurs when a sequence of sensor outputs is detected for which the probability given the sub-goal model is greater than the probability given the toying model.

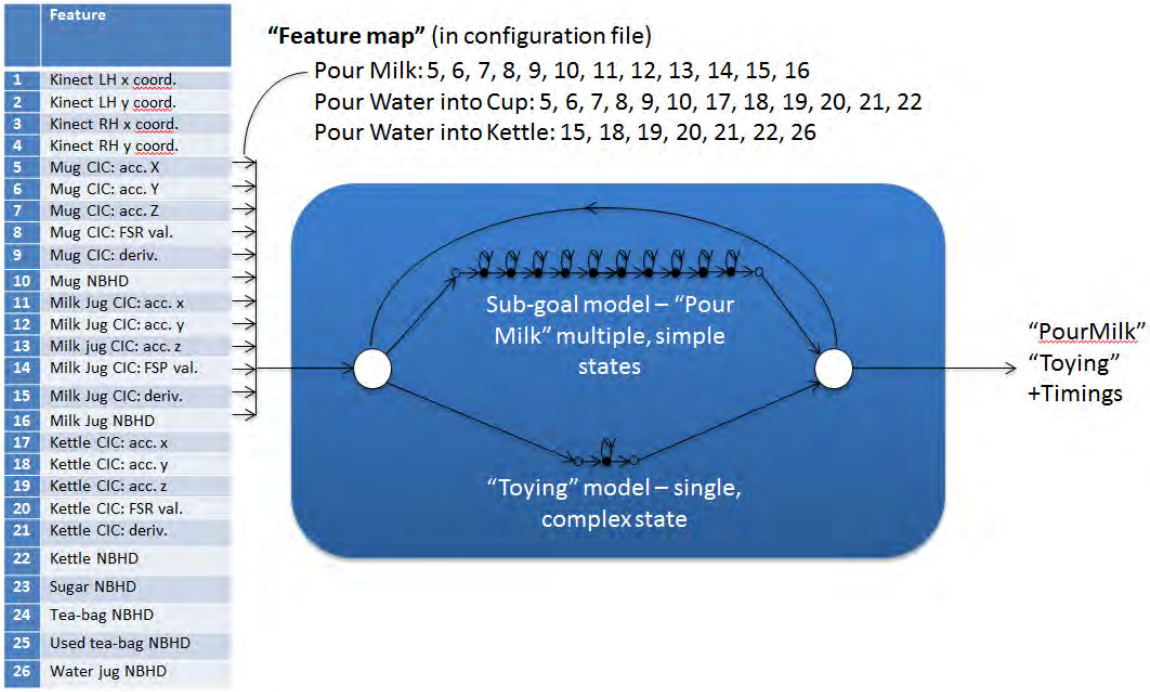


Figure 6: Structure of an individual sub-goal detector

2.2.6 Recognition strategy for the sub-goals of tea-making

The complete sensor set in the CogWatch AR for the prototype tea-making system comprises:

- Two 'standard' CogWatch Instrumented Coasters (CICs). One is attached to the base of the milk jug and one to the base of the mug.
- One 'customized' CIC for the kettle. Because the kettle is a 'base' kettle (i.e. it has a separate base onto which it is normally placed and which is the source of electrical power for boiling, but which it is lifted off for pouring), the standard CIC is not applicable. Hence a customized two-component version of the CIC was developed for the kettle. The three FSRs, which are normally fixed to the base of the CIC, were fixed to the bottom of the kettle base. A separate battery and Bluetooth module are provided for these sensors. The accelerometer, with its own battery and Bluetooth module, is packaged in a small box that is attached to the kettle handle.
- A standard Microsoft Kinect for Windows.

The number of CICs was restricted by the number of devices that needed to be manufactured to provide an identical sensor set for each of the CogWatch partner sites, and the number of devices that can be connected via Bluetooth. This necessitated some compromises in the instrumentation of the task:

- Although the mug, milk-jug and kettle were fitted with CICs, the water jug was not. Therefore components of sub-goals that involve manipulation of the water jug can only be recognized indirectly from Kinect hand-position coordinates or from changes in the weights of other objects (primarily the kettle) that result from actions involving the water jug.

- The tea-bag, used tea-bag and sugar containers were not fitted with CICs, and there are no sensors of any type attached to the individual tea-bags or sugar cubes. This means that, like actions involving the water jug, sub-goals or components of sub-goals that involve manipulation of tea-bags or sugar can only be recognized indirectly from Kinect hand-position coordinates or from changes in the weights of other objects (primarily the mug) that result from actions involving these items.

2.2.7 Real-time recognition

The standard algorithm for HMM-based pattern recognition is the Viterbi decoder. Given a sequence of feature vectors $o_1, o_2,\ldots,o_T$ and a set of HMMs the Viterbi algorithm finds the sequence of models $m_1,\ldots,m_N$ such that (an approximation to) the probability

$p(o_1,\ldots,o_T \mid m_1,\ldots,m_N)$

is maximized. This involves an iterative computation where the probability that the partial sequence $o_1,\ldots,o_t$ is generated and the system is in state i at time t is calculated from corresponding probabilities at time $t-1$ for all possible preceding states j. In the standard implementation of the Viterbi algorithm this continues until the end of the file is reached (t = T) and the optimal explanation of the data can be recovered. For example, this is the basis of the Viterbi decoder “HVite” in the standard HTK toolkit.

However, in a real-time implementation of the Viterbi decoder, data arrives continuously and there is no concept of the ‘end of the sequence’. Without this the algorithm cannot output its decision and eventually the computer’s memory will be exhausted.

2.2.8 Partial Trace-back

A solution to this problem is to use an algorithm called Partial Trace-back. This is illustrated in Figure 7. The partial trace-back algorithm was developed in the context of automatic speech recognition, and the terminology used reflects this. Partial trace-back uses a time-indexed array of word-link records. At any time s in the past, the word-link record at time s contains the identity of the best model that ends at that time, the time in the past when that model was entered, and the corresponding score.

During the Viterbi algorithm, in addition to updating the best score/probability for each state i and time t, a record is also kept of the time at which the model was entered to achieve that score. These are the straight arrows on the right-hand side of Figure 7 and they can be thought of as pointers back into the word-link record.

At regular intervals, these pointers are traced-back through the word-link records to try to find a point in the past, more recent than the start of the buffer, where all of these paths converge (the ‘convergence point’ in Figure 7). Once such a convergence point exists it will always exist, because nothing that happens in the future can change it.Therefore the sequence of words (sub-goals) in portion of the word-link records between the start of the buffer and the convergence point can be output and the corresponding memory can be freed.

2.2.9 Latency

One problem with partial trace-back is that it introduces a time delay, or latency, which is the difference between the time of convergence in the past and the current time. It is important to realize that this latency is not due to a shortage of computing power. Instead it results

from an inability to find a convergence point in the recent past. This is normally because there are multiple, competing explanations of the data which do not converge.

For example, if the final state of the sub-goal model corresponds to a point after completion of the sub-goal where there is no further activity relating to that sub-goal, and if the "toying" model includes a component that corresponds to no activity related to the sub-goal (which it almost certainly will), then after completion of the sub-goal there is potential conflict between paths that remain in the final state of the sub-goal model and paths that remain in that state of the toying model. This type of behavior has been observed and can be fixed by removing the option of remaining in the final state of the sub-goal model by setting its self-transition probability to zero.

Specifically, consider the case of the sub-goal "add milk". The sub-goal HMM is a multi-state model whose states should correspond to the sequence of sensor outputs that typically occur when the sub-goal is executed. During model training, examples of these sensor outputs are automatically aligned with the model. However, a typical training data file for "add milk" will also include "toying" (sensor outputs that follow the sub-goal but are not part of the sub-goal) at the end of the file. If there are errors in the alignment then it is likely that the one or more states at the end of the sub-goal HMM will correspond to "toying" rather than part of the sub-goal. During Action Recognition the sub-goal model competes with the toying (or background) model to explain the sequence of outputs from the sensors. In the situation envisaged above, at the end of an instance of "add milk" the final state of the sub-goal HMM and the "toying" model will provide equally valid interpretations of the data. However, the partial trace-back record for the final state of the sub-goal model will point back to the start of the sub-goal, while the corresponding record for the toying model will point to the end of the sub-goal. Until this ambiguity is resolved (for example, by the patient doing an action which causes sensor outputs that were never seen in the training data after "add milk") the partial trace-back algorithm will be unable to output the sub-goal and there will be a delay. The solution is either to ensure that the end of the sub-goal is accurately labelled in all of the training files (so that the alignment error referred to above never happens) or to edit the sub-goal HMM to remove any states that correspond to toying, or at least to prevent the Viterbi decoder from remaining in these states (as described above).

2.2.10 Implementation of the real-time AR

The real-time CogWatch AR consists of a set of parallel sub-goal detectors, where each detector consists of an implementation of the Viterbi decoder with partial trace-back running on a set of HMMs (comprising at least a sub-goal HMM and a "toying" HMM). The system is implemented in C#.

The object-oriented nature of C# lends itself well to this application. A detector corresponds to a class called "hmmset", for which there are methods corresponding to the Viterbi decoder and the partial trace-back algorithm, and the whole AR is just a set of detectors.

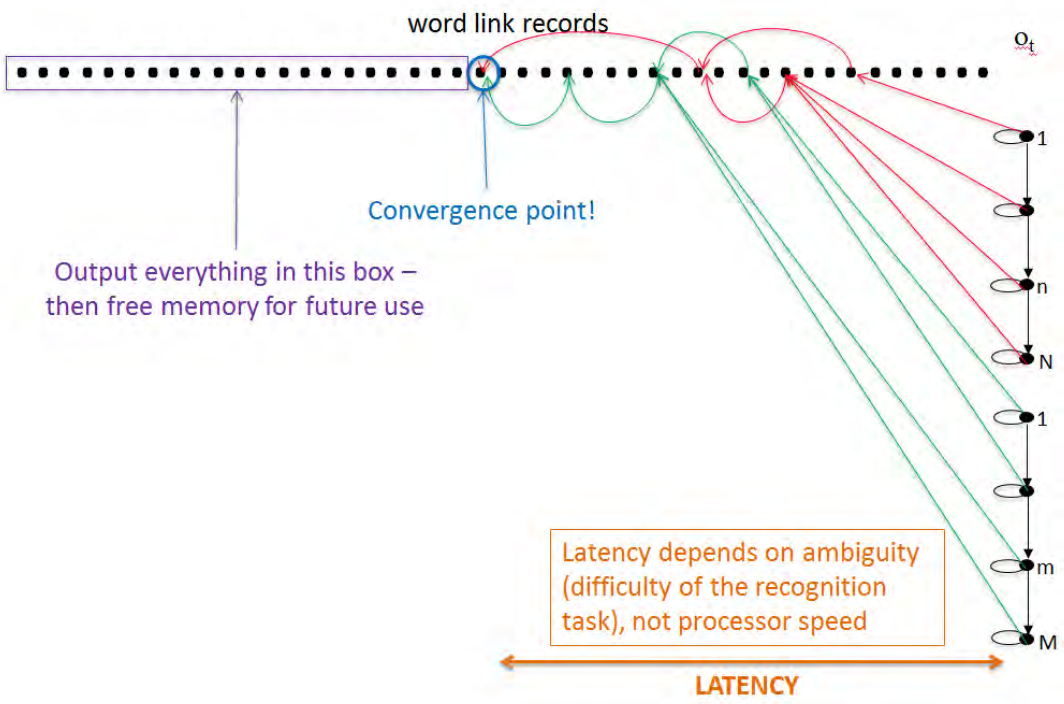


Figure 7: Explanation of the Partial Traceback algorithm to enable real-time Viterbi decoding.

2.2.11 Validation

The CogWatch AR uses exactly the same file formats to store HMMs as the standard HTK system. This has two major advantages. First, the parameters of the sub-goal and toying models can be learnt automatically from data using the standard HTK model parameter estimation tools. The models can then be ported to the CogWatch AR. Second, the results of the detailed calculations performed inside the CogWatch AR can be compared with those performed inside the HTK tool HVite. In this way the correct operation of the CogWatch AR has been validated.

2.3 Off-line experimental evaluation

This section reports the results of an off-line experimental evaluation of the performance of the individual sub-goal detectors in the current version of the CogWatch AR.

2.3.1 Data used in the evaluation

Twenty-six participants, aged between 18 and 80, completed multiple individual sub-goals and multiple full tea-making trials. In all cases synchronized sensor outputs were recorded.

The recordings are summarized in Table 1: Details of recordings of data used in the experiments reported in this section. In total there are 1124 recordings of isolated actions (4.01 hours) and 99 recordings of complete tea-making (2.51 hours).

Sub-goal	Trials	Duration (hours)
Pour kettle	138	0.64
Add milk	123	0.41
Add sugar	120	0.34
Add teabag	144	0.31
Fill kettle	146	0.71
Remove teabag	134	0.63
Stir	138	0.56
Toy(1)	26	0.07
Boil water	125	0.22
Toy(2)	30	0.11
Full trial	99	2.51

Table 1: Details of recordings of data used in the experiments reported in this section.

2.3.2 Effect of combining sensor-based and Kinect-based features

Figure 8 shows recognition accuracy for the sub-goal “fill kettle” using only the data from the sensors in the kettle CIC (left) and the CIC information plus Kinect hand-coordinates (right). In both graphs the horizontal axis shows the results of varying the number of Gaussian components in the “toying” model, while the different colored curves correspond to different numbers of states in the sub-goal (“fill kettle”) HMM.

The left-hand graph in Figure 8 shows that with 8 or more Gaussian components in the “toying” model and up to 5 states in the sub-goal model, a recognition accuracy of 90% can be achieved. As this result does not use hand coordinate data from Kinect, and the water jug does not have a CIC, this result relies entirely on changes in the weight of the kettle. The fact that this is the only useful cue to the “fill kettle” sub-goal also explains why so few states are needed in the sub-goal model. The right-hand graph in Figure 8 shows the corresponding results when hand-coordinate data from Kinect is included. In this case, the accuracy increases to over 95%. In other words, the inclusion of the Kinect hand-coordinate data through the mechanism of Gaussian neighborhoods reduces the error rate by over 50%.

Further improvements have been made to the sub-goal model for “fill kettle” so that recognition accuracy is now close to 100% using the kettle CIC and Kinect. Performance is similarly high for the other sub-goals where significant use can be made of the CIC sensors. For example, recognition accuracy for “pour kettle” and “add milk” are currently 98.5% and 99.6% respectively.

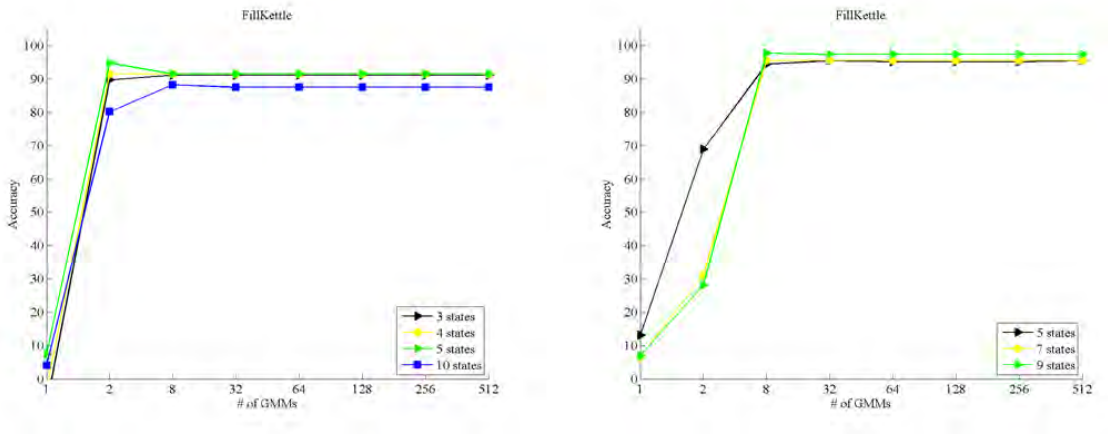


Figure 8: Comparison of recognition accuracy for the sub-goal “fill kettle” using just CIC sensor information (left) and CIC sensor information plus Kinect hand coordinates (right).

2.3.3 Recognition of “add tea-bag”, “add sugar” and “remove tea-bag”

The “add tea-bag”, “add sugar” and “remove tea-bag” sub-goals are referred to as front actions because the containers are located at the front of the table. These are the most challenging sub-goals to recognize because the containers are not fitted with sensors and it is not possible to directly instrument a tea-bag, used tea-bag or sugar cube. Recognition relies entirely on Kinect hand coordinate data and very small changes in the weight of the mug.

Figure 9 shows the result of a recognition experiment to detect the three front actions using only hand coordinate data from Kinect together with the Gaussian neighborhoods for the tea-bag container, sugar container, used tea-bag container and the mug. This experiment differs from the previous one in that a single detector is being used for all of these sub-goals. In other words the sub-goal detector contains three sub-goal HMMs (for “add tea-bag”, “add sugar” and “remove tea-bag”) plus a single “toying” model. The figure shows that with a “toying” model with at least 8 Gaussian components error rates of between 20% and 30% can be achieved. It is interesting to note that the best performance is achieved with small sub-goal models (just three states).

Figure 10 shows the results of the same experiment but using the outputs of the FSRs in the mug CIC in addition to the Kinect hand coordinate data. Including the FSR data reduces the error rate to between 10% and 20%. An interesting difference between this and the previous result is that the best performance is now obtained with the bigger sub-goal HMMs (10 states) and the smallest model gives the poorest result. Presumably this is because the sequential structure of the signal becomes more complex when the FSR data is added.

Figure 21 shows the results of the same experiment but using Kinect hand (x,y) coordinates plus their derivatives, and the FSR outputs from the mug, milk jug and kettle CICs. The derivatives of the hand coordinates allow the system to distinguish between hand

movements that pass through the relevant Gaussian neighborhoods and those which pause in these neighborhoods, as would be the case for adding a tea-bag or sugar or removing a used tea-bag. The relevance of the FSR outputs for the milk jug and kettle is that these should be constant, and if they are not then the user is probably doing something other than one of the front actions.

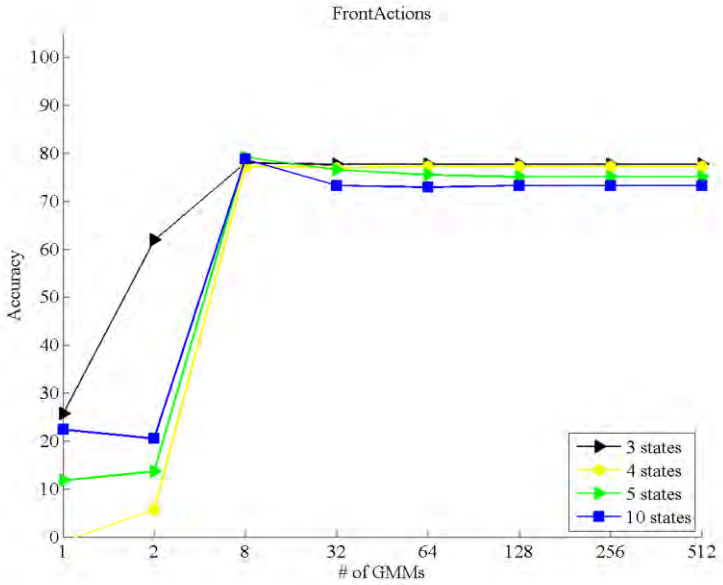


Figure 9: Recognition accuracy for the three “front actions” using Kinect hand coordinate data only.

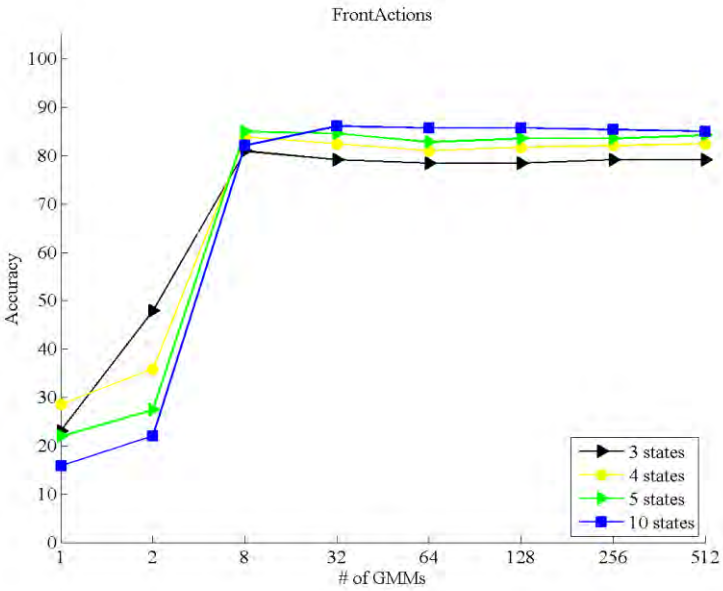


Figure 10: Recognition accuracy for the three “front actions” using Kinect hand coordinate data plus the outputs of the FSRs in the mug CIC.

The use of these additional sensor inputs reduces the error rate for the front actions to less than 10%, provided that the sub-goal models are sufficiently long (more than 4 states) and the toying model is sufficiently complex (more than 32 Gaussian components).

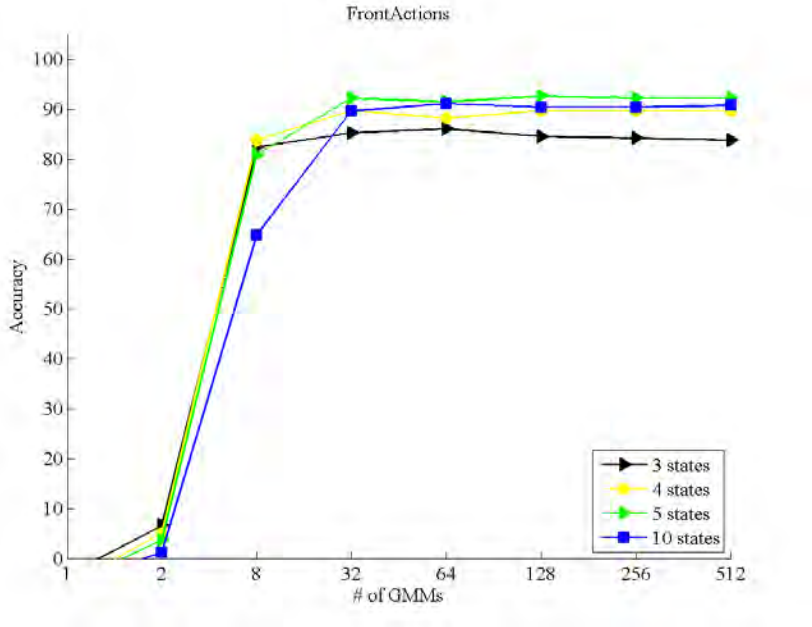


Figure 21: Recognition accuracy for the three “front actions” using Kinect hand coordinate data and derivatives, plus the outputs of the FSRs in the mug, milk jug and kettle CICs.

2.3.4 Effect of 50Hz sample rate

In the original version of the CogWatch system the sensor outputs were sampled at 200Hz. However, due to problems with the reliability of the Bluetooth connection the sample rate was reduced to 50Hz. Action recognition experiments were conducted using 200Hz, 100Hz and 50Hz sample rate sensor data to measure the effect, if any, on recognition accuracy of changing the sample rate.

The experiments confirmed that reducing the sensor data sample rate from 200Hz to 50Hz has no significant effect on recognition accuracy.

2.4 Summary of real-time Action Recognition for tea-making

2.4.1 AR System

A real-time AR system has been implemented for the CogWatch project. The system uses hidden Markov models (HMM) to model the sequences of sensor data that correspond to the various sub-goals of the tea-making task. The real-time decoder is a Viterbi decoder with partial traceback and is implemented in C#.

The HMMs that are used in the system are trained on recordings of real tea-making trials using the standard Cambridge University Engineering Department HTK HMM toolkit. The model formats used in HTK are also used in the CogWatch system.

The CogWatch AR has been validated against the HTK Viterbi decoder tool HVite and has been shown to give identical results, given the same HMM set and data.

2.4.2 AR Performance

The performance of the AR system has been measured in off-line experiments using HTK. The results show that for sub-goals where all of the relevant objects are instrumented with a CIC (such as “add milk” or “add (boiling) water (to the mug)”) very high recognition accuracy can be achieved. For sub-goals where only a subset of the objects are instrumented (such as “fill kettle”, where the kettle is instrumented but the water jug is not) error rates can be reduced by 50% by including hand coordinate data measured using the Microsoft Kinect system.

For sub-goals where the objects have minimal instrumentation (for example the “front-action” sub-goals “add tea-bag”, “add sugar” and “remove tea-bag”), an error rate of less than 10% has been achieved by suitable processing of the Kinect hand coordinate data and incorporation of sensors that are not directly involved in the sub-goal. For example, if the user’s hand follows a typical trajectory for “add tea-bag” but the FSRs attached to the base of the kettle show that its weight is increasing, then “add tea-bag” would be an error and the evidence points towards “fill kettle”.

2.4.3 Real-time issues

The system is able to run a full set of 10 sub-goal detectors in parallel in real-time on a standard PC without any issues arising relating to computational load. The main challenge for real-time operation has been to remove the delay that is a consequence of the partial trace-back algorithm. A simple solution is to set the self-transition probability for the final state of the sub-goal HMM to zero. This compensates for the situation where the PDF associated with this state may be identical to one of the components of the “toying” model.

2.4.4 Future Work

Current work is focusing on three main areas:

- Data collection: Recordings of the sensor outputs that are obtained when users execute individual sub-goals and whole tea-making tasks are being collected at UOB, TUM and UPM. These will be used to train the parameters of an improved set of sub-goal and toying HMMs.
- Task and sub-task level HMMs: Recall that a sub-goal can be further split into tasks and sub-tasks. This is analogous to splitting a word into phonemes in automatic speech recognition, and hence all of the relevant tools already exist in HTK (because this is the standard approach in speech recognition). Pilot experiments have shown that the decomposition of sub-goals into tasks and sub-tasks can improve recognition accuracy. For example, in the sub-goal “add milk” or “pour kettle” an explicit sub-task model for “tilt jug (or kettle)” ensures that this critical action is part of the sub-goal model and is not overlooked. A minor upgrade to the real-time CogWatch AR will be needed to accommodate the partition of sub-goals into tasks and sub-tasks.
- Latency: A consistent and principled approach to reducing latency of the partial-trace-back algorithm needs to be developed. It is believed that this will be best achieved through the explicit partition of sub-goals into tasks and sub-tasks. For

example, the “rest” period at the end of a sub-goal could be modelled explicitly and not be part of the sub-goal model itself.

2.5 Exploitation

The CogWatch consortium appreciates the need to exploit the technology that has been developed under CogWatch, and considerable thought has been given to the exploitation of the real-time AR.

The principles that underpin the algorithms inside the AR are not new. The value of the real-time AR system stems from its embodiment of “know-how” rather than anything that is patentable.

It has already been explained that the CogWatch AR is compatible with the standard Cambridge University HTK system. This is used in hundreds of laboratories worldwide to build experimental HMM-based speech recognition systems. Increasingly it is also being applied to other applications, because it is generic and can be used to model and detect patterns in any sequential data. HTK includes a tool for off-line Viterbi decoding (HVite) but there is no compact tool for real-time decoding that can also support multiple, parallel decoders (like the CogWatch AR can).

It has been decided that the best way to obtain maximum benefit from the effort that has been invested in the CogWatch AR is to share it as a free download via a site such as GitHub. This will require some improvement of the basic software, which will be done. The standard license used by GitHub stipulates that any software that is developed using the CogWatch AR would also need to be shared through the same site. Any user that is unwilling to follow this arrangement would be required to negotiate directly with the developers of the software (UOB) for a commercial license.

3. SEGMENTATION AND KINEMATIC-BASED RECOGNITION OF TEA MAKING IN HEALTHY SUBJECTS UND PATIENTS WITH ADL DEFICITS

3.1 Introduction

This part of the deliverable reports results from experiments investigating the potential of kinematic analyses to support action recognition (AR) in the Cogwatch systems. Likewise, the kinematic characteristics of the multi-step action of tea making are described in healthy subjects and in patients with ADL deficits following stroke.

The decomposition of selected activities of daily living, defined as action goals, into the constituent sub-actions, defined as sub-goals, is an essential part of the Cogwatch system's architecture. The task of the activity recognizer is to identify the current sub-goal on-line from sensor signal acquired by the coasters and by the Kinect. As outlined in the present and previous Cogwatch reports, Hidden Markov Models have been implemented for the purpose of activity recognition. This statistical approach has a limited precision and depending on various factors, decisions may be ambiguous (see above). Information about the kinematics of hand movements may be beneficial to increase the precision and decrease ambiguity of activity recognition.

In a previous study we found that kinematic information of the hand movements can be sufficient to detect and identify the sub-actions of the tea making task (see Deliverable D 3.1). However, when the sub-actions are executed with variable order and in particular when stroke patients with slowed movement execution are investigated the precision of this method decreases substantially. Therefore we evaluated the combined use of the signals from the Cogwatch instrumented coasters and from hand kinematics (Hughes, Parekh, & Hermsdörfer, 2013). The processing involved two steps. Characteristic events in the coaster signals identified the sub-action and marked its initial and the final phase. Backward-search and forward-search for local minima in the absolute velocity profile then served to locate the exact beginning and the end of the sub-action. This method enabled to identify and extract the sub-actions or action segments from the stream of data. The accuracy of the approach, as assessed by comparison with video recordings, was between 60 and 90 % of correctly identified and extracted action segments. The accuracy depended strongly on whether the sub-actions yielded characteristic sensor signals. Another limiting factor was the quality of the coaster data that might become less of a problem due to change from 200 Hz to 50 Hz coaster data transmission with fewer transmission errors (see elsewhere in this report).

The present report on hand kinematics did not involve sub-action identification using the coaster signals. Rather the segments were identified using video recordings and kinematics was used for fine adjustments of start and end. Thus the results always relate to the correct action segment which was considered a prerequisite for the present analysis. One has to be aware however, that an application of kinematic data in the Cogwatch system demands online automatized segment detection as suggested by Hughes et al. (2013).

To assess the kinematic characteristics of hand movements dependent on the sub action, the particular hand used, and the presence of AADS impairments, we first calculated various kinematic measures for the movements of both hands in nine healthy subjects and in seven stroke (CVA) patients. Depending on characteristics of the individual lesion, patients may be able to use both hands in their daily activities or they may have to rely on

their ipsilesional hand when the function of contralesional hand is absent due to plegia. Therefore tea making was assessed with all possible combinations of bimanual and unimanual hand use.

3.2 Materials and Methods

3.2.1 Task and Procedure

In the version of the tea-making task tested here, participants are instructed to prepare a cup of tea with milk and one sugar cube. Thus following items were placed on the table: kettle, teabags, milk, sugar cubes and an additional distractor item (instant coffee jar).

Following conditions were tested:

- bimanual: use of both hands
- unimanual: use of the ipsilesional hand in patients and the dominant hand in healthy subjects, respectively
- unimanual: use of the contralesional hand in patients and the non-dominant hand in healthy subjects, respectively

Every condition was repeated once, resulting in a total of six trials. The order was bimanual, unimanual (ipsilesional / dominant), unimanual (contralesional / non-dominant).

The settings of the objects available in the task are shown in Figure 12. Starting positions for the left and the right hand are represented by the labeled papers. In the beginning of each trial, the water jug is filled with approximately 0.5 liters of preheated water, the milk jug is filled, the teabag labels are prepared for an easy entanglement, particular in unimanual trials, and the kettle body is empty. The containers handles are directed towards the subject.

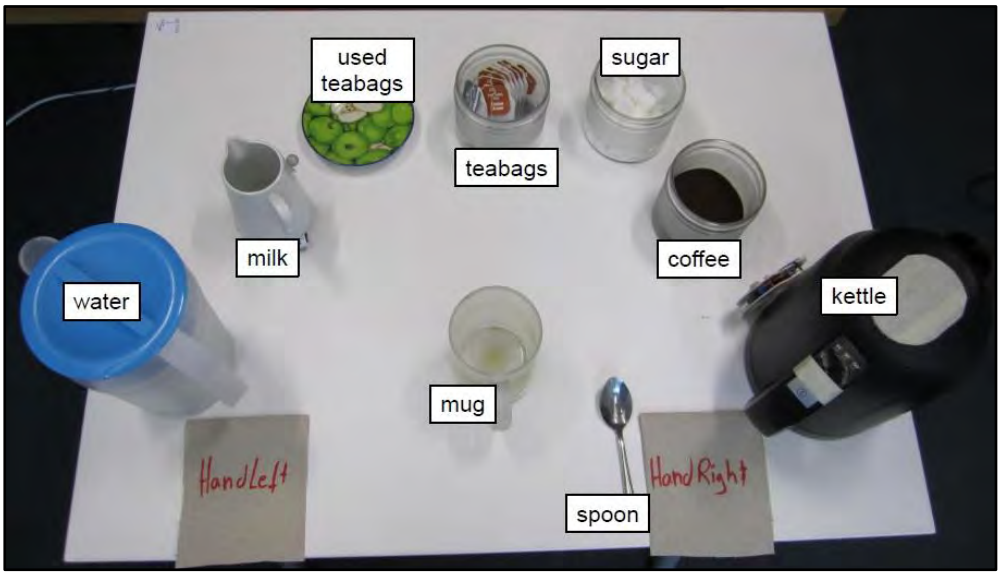


Figure 12: Experimental setting for the tea-making task including a water jug, milk, a plate for used teabags, teabags, sugar, coffee, a kettle, a mug and a spoon.

Until now 9 controls and 7 CVA patients (3 with left brain and 4 with right brain damage) were tested and analyzed, 67 trials in total, of which 41 were performed by controls, 16 by patients with right brain damage (RBD) and 10 by patients with left brain damage (LBD). Patients were recruited from the Clinic for Neuropsychology at the Hospital München-Bogenhausen in Munich. Patient's age ranges from 47 to 79 years with a mean of 63 (±9.1 y) years and time since stroke between 0.5 and 6.5 years with a mean of 2.5 (±2.2 y) years. Controls have a mean age of 70.9 years (±3.4 y). One of the LBD patients, four of the RBD patients and three of the control subjects were male. Subjects were tested for the handedness by the Edinburgh Handedness Inventory (EHI). All CVA patients but one LBD patient and all controls subjects but one were right handers, almost all of them strong (13).

Code	**Age**	**Sex**	**Side of Brain Damage**	**Paresis**	**Time since Stroke**	**EHI (%)**
S20	47	M	Left	Yes	1y	100
S22	70	W	Left	Yes	0.5y	100
S36	63	W	Left	Yes	0.5y	0
S85	70	M	Right	Yes	6.5y	68
S93	58	W	Right	Yes	3y	100
S96	79	M	Right	Yes	4y	100
S115	64	M	Right	Yes	2y	80
Mean	*63 ± 8.33*				*2.5 ± 2.2*	

Table 2: Demographic and clinical data of patients tested in the tea-making task.

Subjects are asked to wear a SMI-ETG eye tracking device during task performance. The eye tracking glasses incorporate a HD scene camera with a sampling rate of 30Hz. Fixations are identified and assigned to fixated objects off-line.

Positional data of both hands are recorded with the use of 5 Oqus Motion Capture cameras included in a Qualysis motion capturing system with a sampling frequency of 120 Hz, three passive markers were attached to each hand in the mid-palm section. For the analysis only one marker was used, the additional two were attached for a better recording reliability and in case one or two markers get lost, e.g. by scratching of the subject.

The mug, the milk jug and the kettle's base and body have force and acceleration sensors with a sampling rate of 200Hz (now 50Hz, see elsewhere in this report) attached. These instrumented coasters are custom made by UOB.

3.2.2 Segmentation

In a first step data streams are synchronized using MatLab and a Visual C executable file. The coarse boundaries of the action segments are manually defined via the SMI-ETG HD scene camera's video data. The fine adjustment of action-segments is then performed with the use of hand kinematics in MatLab. Figure 13 shows an example of the data

synchronization with action segmentation of the tea-making task, performed by a CVA patient in a bimanual trial. In the upper row fixated objects are indicated using a color code (see legend on the right side of Figure 13). Note the dominance of kettle fixations in the first and mug fixations in the second half of the trial following the demands of waiting, pouring and monitoring the mug's filling level. The lower two lines (yellow and green, baseline at '0') show the tangential velocities of right and left hand movements with a more intense activity in the first 3 sub-segments of the task. The graphs in the mid-section are the most relevant data of the objects sensors. Note the boiling of the water in the blue kettle acceleration data and impact of the rotation when pouring the heated water into the mug which can be also seen in the orange graph, displaying the force sensors of the mug. The waiting time until the water in the kettle is boiling is cut out from the data stream as indicated by the thick vertical lines. The resulting duration of the task execution is 132s. The numbers refer to the identified sub-segments. Segment 5, removing the teabag, is missing and the sequence of segments 6 and 7, adding milk and sugar, is inverted. Deplorably, such complete sensor data is the exception, so the entire action segmentation process is still done manually.

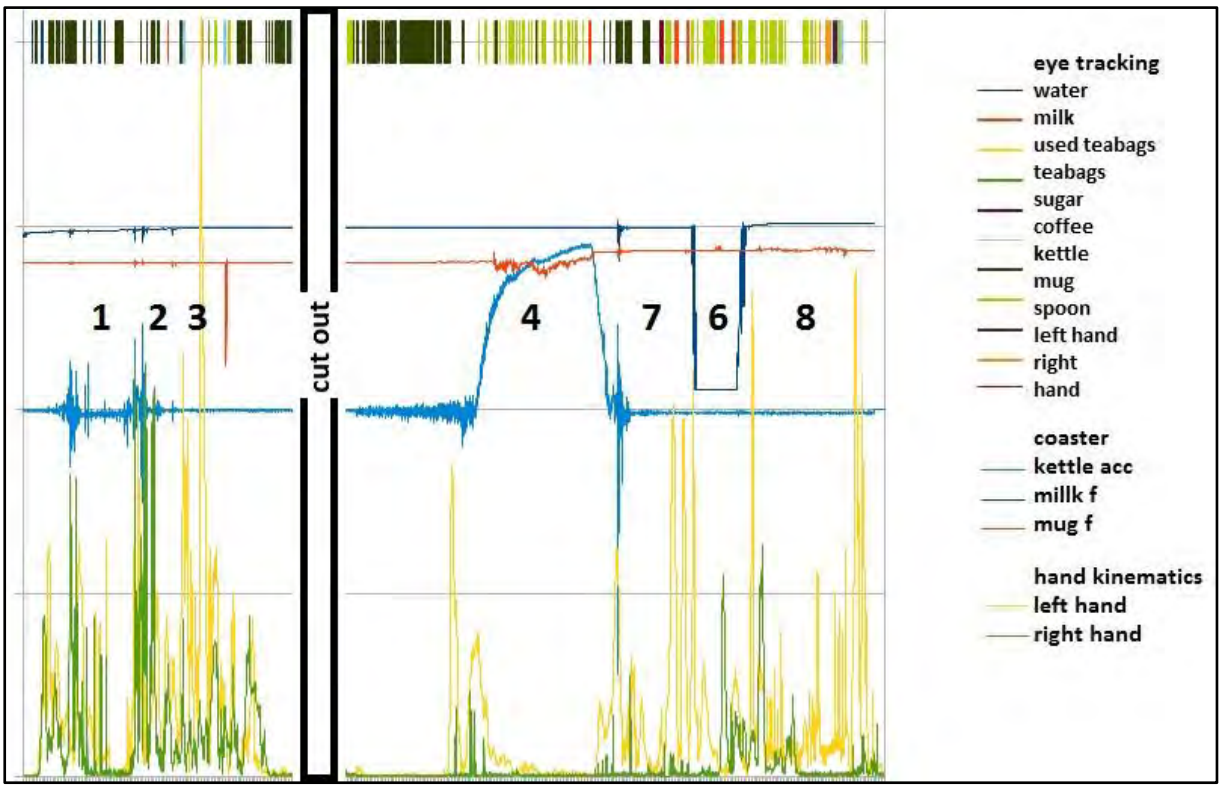


Figure 13: Example of the synchronization of data streams with action segmentation of the tea-making task (bimanual).

After being synchronized, data are segmented into discrete actions and analyzed. The whole task is segmented into the following eight action segments (Humphreys & Forde, 1998 in Forde et al., 2010):

1. pour water in the kettle
2. switch the kettle on
3. place a teabag in the mug
4. pour heated water into the mug
5. remove the teabag
6. add milk
7. add one sugar cube
8. stir the tea

Figure 14 shows an example of action segmentation for one trial performed by a patient. Figure 15 shows the number of executed sub-segments per trial for the control group and the patient groups with left and right brain damage as well as and the frequency of occurrence for the different sub-segments per trial. Note that segment 5 and 8 are frequently omitted, especially in the patient group with right brain damage. The number of executed sub-segments per trial is mainly influenced by sequence omissions.

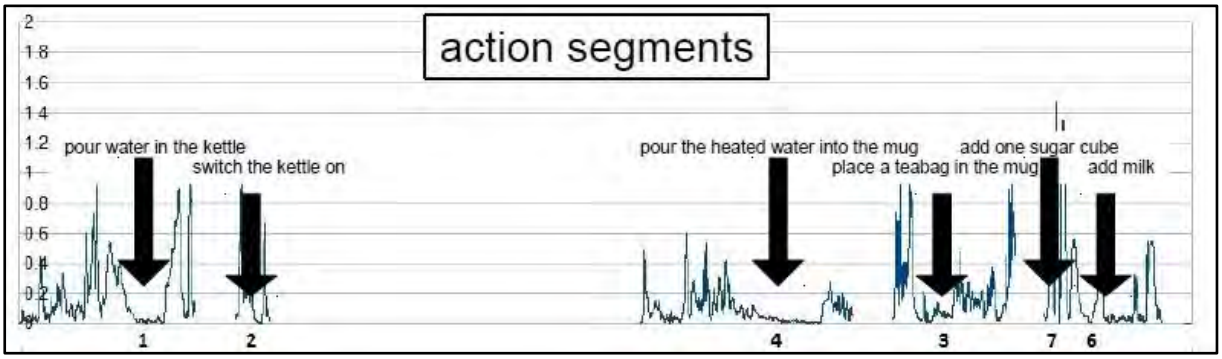


Figure 14: Hand velocity in m/s and segments identified for a patients' trial. Note that only segments 1, 2, 3, 4, 6 & 7 were executed and their order was mixed.

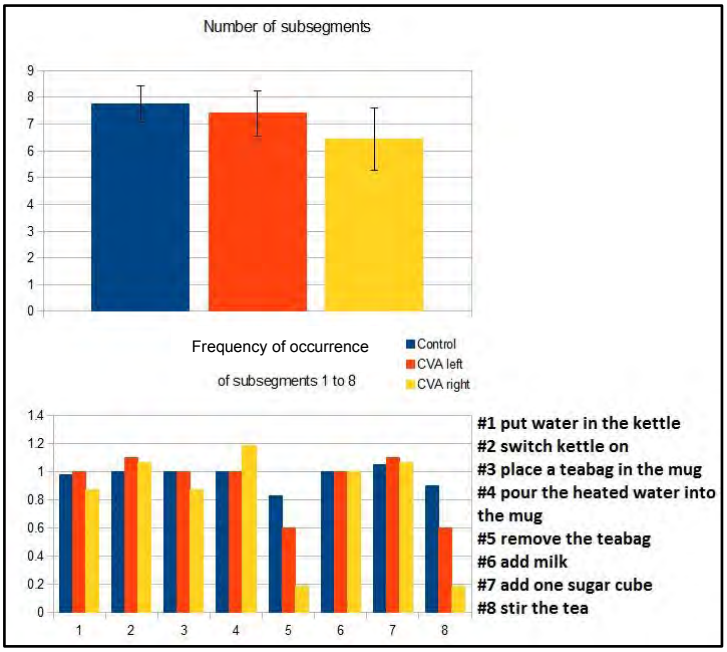


Figure 15: Number of sub-segments performed per trial and average frequency of occurrence of the different sub-segments in the control group and the patient groups (LBD, RBD).

3.2.3 Kinematic analysis

Positions and velocities of the hands are determined from the motion recording and smoothed using a 1s LOESS filter ('local regression'). Following measures are determined for the complete action and action segments:

- maximum peak velocity
- movement times
- path lengths

'Maximum peak velocity' describes the maximum tangential speed reached in the segment respectively in in the whole trial.

'Movement times' defines the time to complete the sub-segments respectively the whole task without the waiting period for the boiling of the water which is usually distinguished by resting hands.

'Path lengths' is the tangential distance traveled by the single hands. Measured path lengths can be increased due to additional action as well as non-goal-directed movements, changes of directions or even tremors.

3.3 Results

Performance of healthy control subjects

The overall movement times for the different task conditions, bimanual, unimanual right and unimanual left, do not differ essentially in the control group (Figure 16). They need a little more than a minute to prepare the demanded cup of tea, time to heat the water excluded.

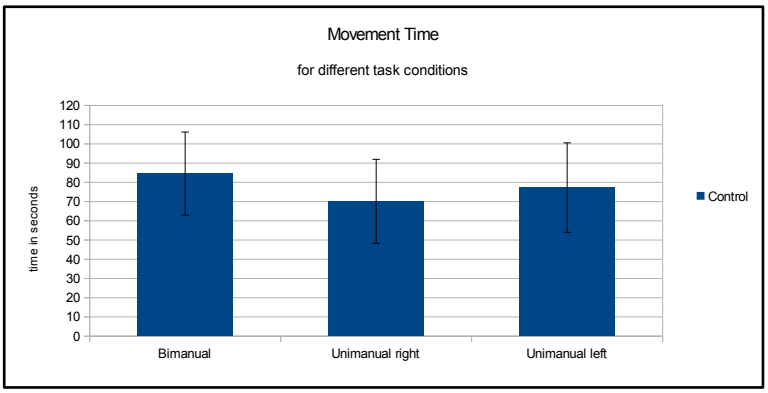


Figure 16: Movement time in total for the different task conditions in controls.

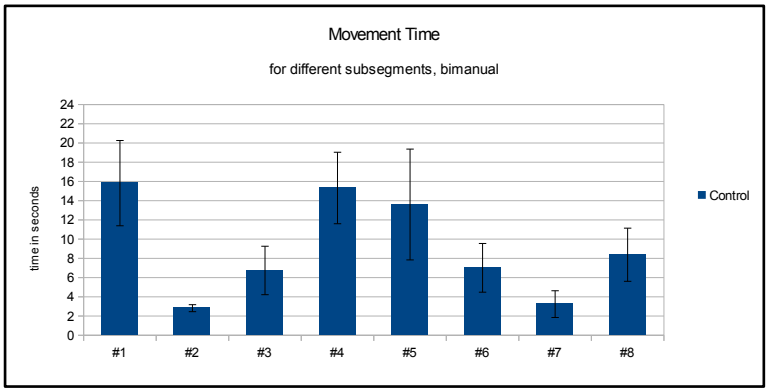


Figure 17: Movement time for the different sub-segments in the bimanual task condition in controls.

Looking at the sub-segments of the bimanual task condition the control group needed very different time periods to execute the subtasks. Segments 1 (pour water in the kettle), 4 (pour heated water into the mug) and 5 (remove the teabag) are the most time demanding sub-segments (Figure 17).

The path length travelled by the hand(s) did not differ between the unimanual task conditions but for the bimanual condition (Figure 18). There, the right hand and particularly the left hand have a shorter path than the hands in the unimanual condition. The sum of the distance of both hands during bimanual execution is however clearly larger than for one hand under unimanual conditions. Thus path length seems not strictly optimized under bimanual conditions. This may be due to enable better control in subtasks like pouring liquids or placing the teabag and / or to avoid awkward positions and movements during the task.

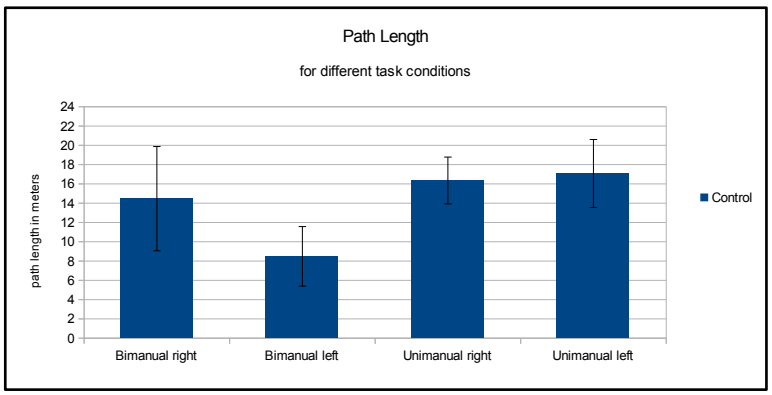


Figure 18: Overall path length for the different task conditions in controls.

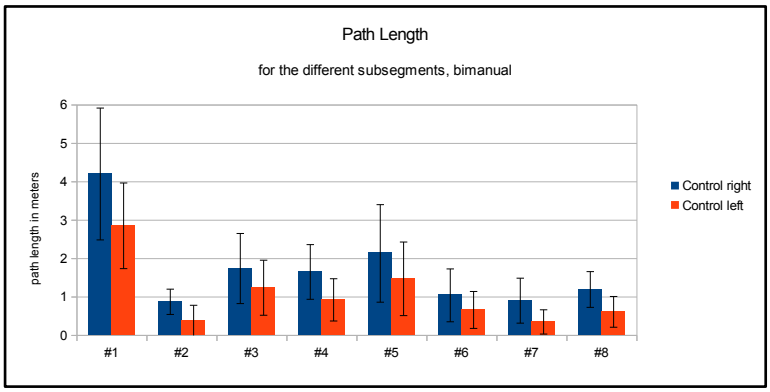


Figure 19: Path length for the different sub-segments in the bimanual task condition in controls.

As displayed in Figure 19, path lengths for the different sub-segments cover a wide range. Interestingly, the ratio between the path length of the right hand and the left hand is not changing that much (1.82 ± .44), which speaks against the hypothesis of better control in subtasks.

The maximum velocity peak in the control group interestingly depends on the task conditions, with the bimanual condition showing the highest velocity peak for the right hand (Figure 20). The assisting left hand in the bimanual condition shows the lowest peak velocity and both unimanual conditions do not differ significantly in the their peak velocities.

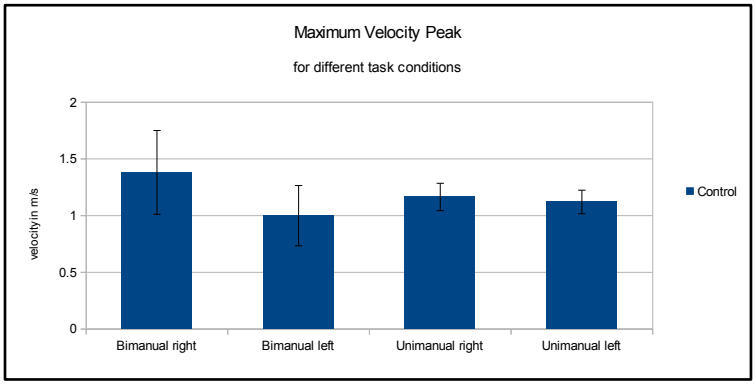


Figure 20: Maximum velocity peak in total for the different task conditions in controls.

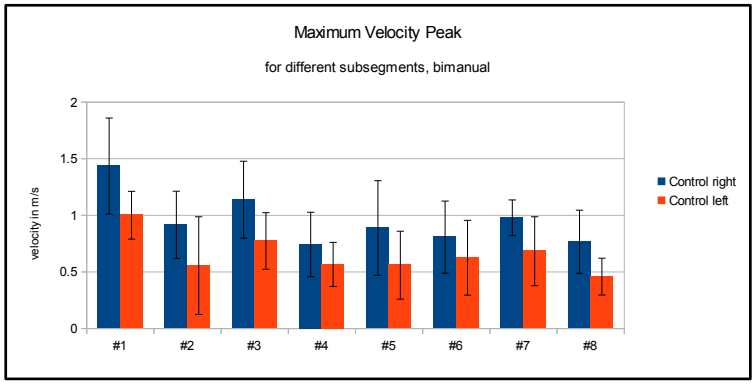


Figure 21: Maximum velocity peak for the different sub-segments in the bimanual task condition in controls.

The maximum velocity peak in the sub-segment in the bimanual task condition is highest for segments including large reaching or transporting movements like for the replacement of the water carafe or the reaching for the teabags, while it is smallest in segments with almost only pouring actions and short reaching movements like the adding of the milk (Figure 21).

Performance of patients with brain damage

Movement Time

Comparing overall movement duration for the different task conditions between the control group and the two patient groups, both patient groups show significantly higher movement times for the task (Figure 22). Note that patients had longer movement times in the bimanual task condition than in the corresponding unimanual trials. This might be due to the demands of intermanual coordination. While none of the patients with left brain damage was able to perform the task solely with the contralesional hand, all of the 4 patients with right brain damage were able to do so. These patients show fastest task execution in the contralesional condition. This unexpected outcome might be due to additional training with their impaired hand.

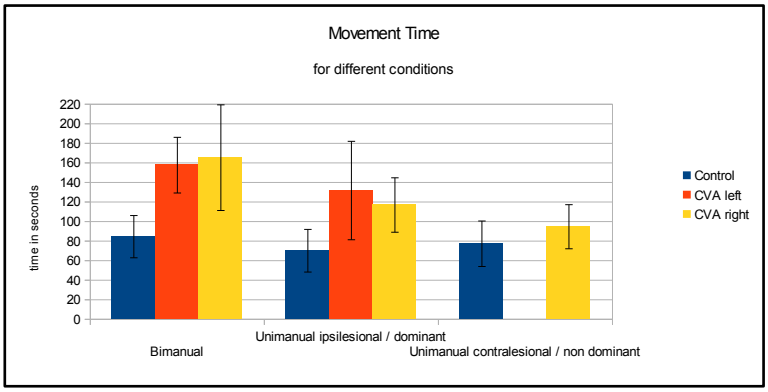
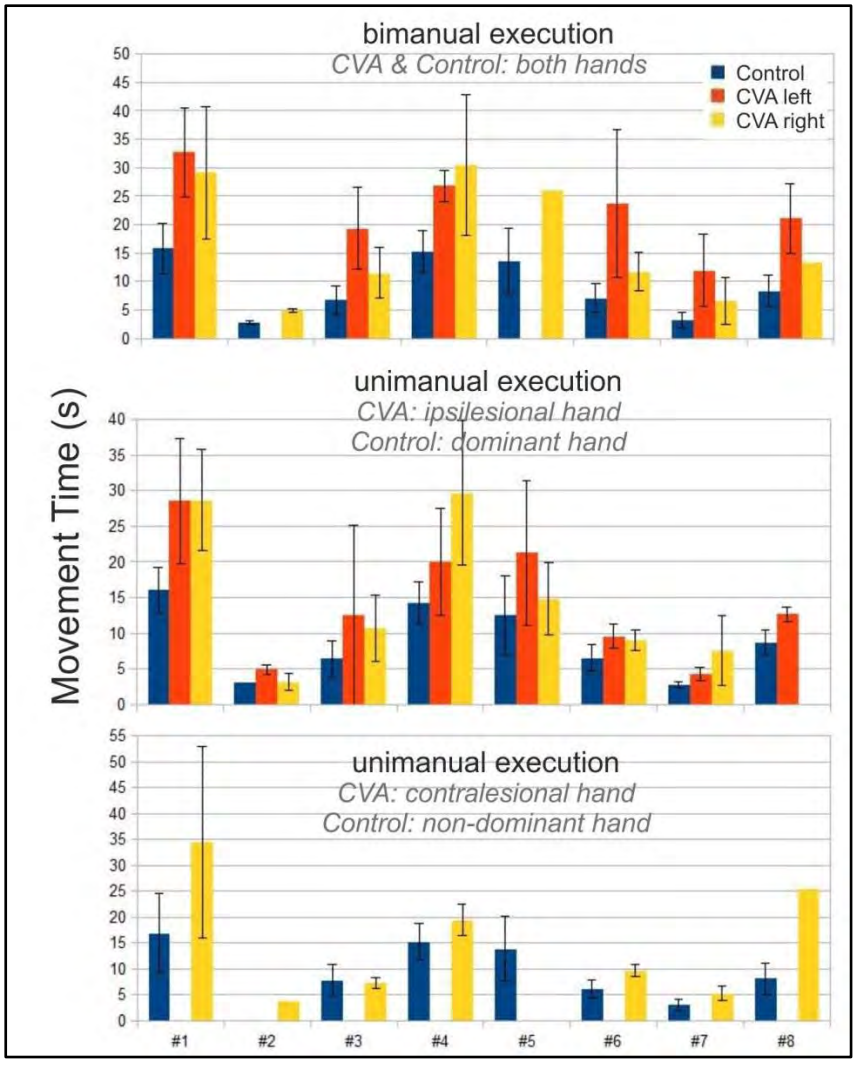


Figure 22: Movement time in total for the different task conditions in controls, patients with left brain damage and patients with right brain damage.



.

Figure 23: Movement time for the different sub-segments in the three task conditions in controls, LBD & RBD patients.

The movement times for the different sub-segments in the bimanual condition reveal a comparable tendency as for the overall movement times when comparing the groups: Patients need more time for the subtasks. Especially patients with left brain damage show extreme variations of the time used for single sub-segments, while patients with right brain damage in general need less time in the sub-segments. This seems surprising since their overall movement time is even higher than that of the patients with left brain damage (Figure 22 & Figure 23). This might be due to longer resting periods between the segments, maybe because of sequencing problems

In the ipsilesional hand condition the patient groups again show longer movement times for the different sub-segments, but the LBD patients do not show the extreme timespans like in the bimanual condition (Figure 23). Differences between the patient groups appear in segments 3, 4, 5, 6 and 7. Patients with right brain damage seem to have problems with pouring the heated water into mug and adding a sugar cube, while patients with left brain damage need longer to place and remove the teabag from the mug and adding the milk. In the unimanual non-dominant / contralesional hand condition movement times for segments 1 and 8 are very high in the patient group with right brain damage and overall higher (except segment 3) than in the control group (Figure 23).

Path Length

The overall path lengths for the different task conditions are quite similar for the groups (Figure 24). RBD patients reveal a similar contribution of path length across hands as controls. This can be expected since the right hand is typically unaffected by paresis in this patients. But interestingly the path for LBD patients is longer for their contralesional right hand than for their ipsilesional left hand. Apparently they act as slight right handers despite their left brain damage and right-sided paresis.

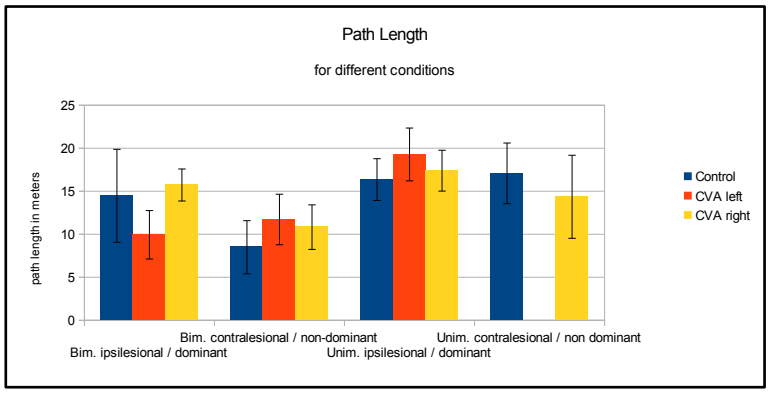


Figure 24: Path length in total for the different task conditions in controls, patients with left brain damage and patients with right brain damage.

Figure 25 shows the path lengths for the different sub-segments in the three task conditions for the groups. Although, in the bimanual task condition, there seem to be big differences in the path lengths of the different subtasks, standard deviations are high and overall control and patient groups do not really differ. Similarly, path lengths do not differ between groups in any other condition (Figure 25). Obviously on a sub-segment level, the distance travelled by the hands does not seem to be affected by the stroke in the patient groups.

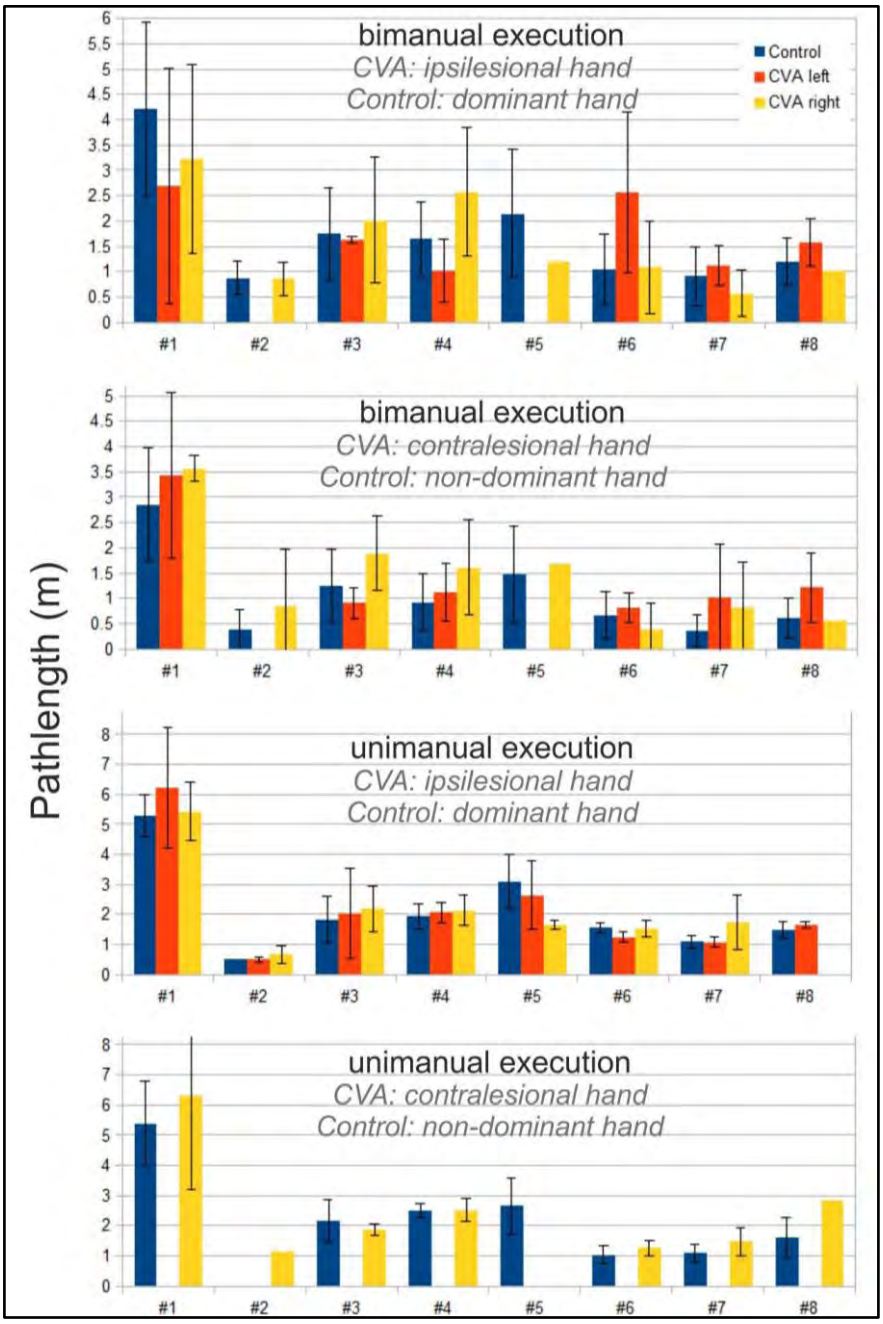


Figure 25: Path lengths for the different sub-segments in the three task conditions in controls, LBD & RBD patients.

Maximum Peak Velocity

The overall maximum velocity peak in the different task conditions is lower for the patient groups, especially for the LBD patients (Figure 26). This fits quite well with the preceding observations, where patients showed longer movement times and comparable path lengths.

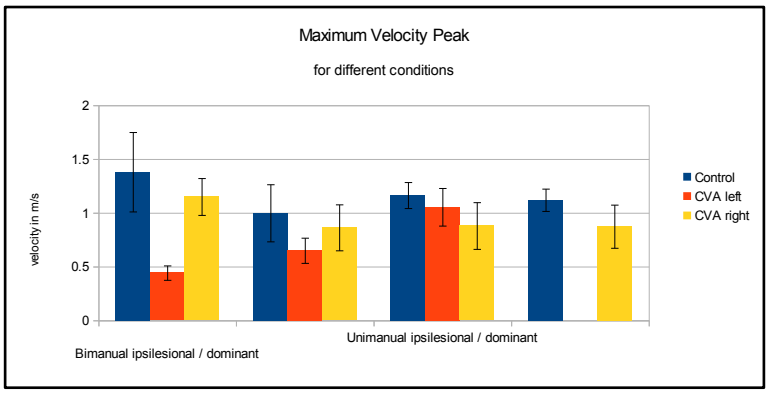


Figure 26: Maximum velocity peak in total for the different task conditions in controls, patients with left brain damage and patients with right brain damage.

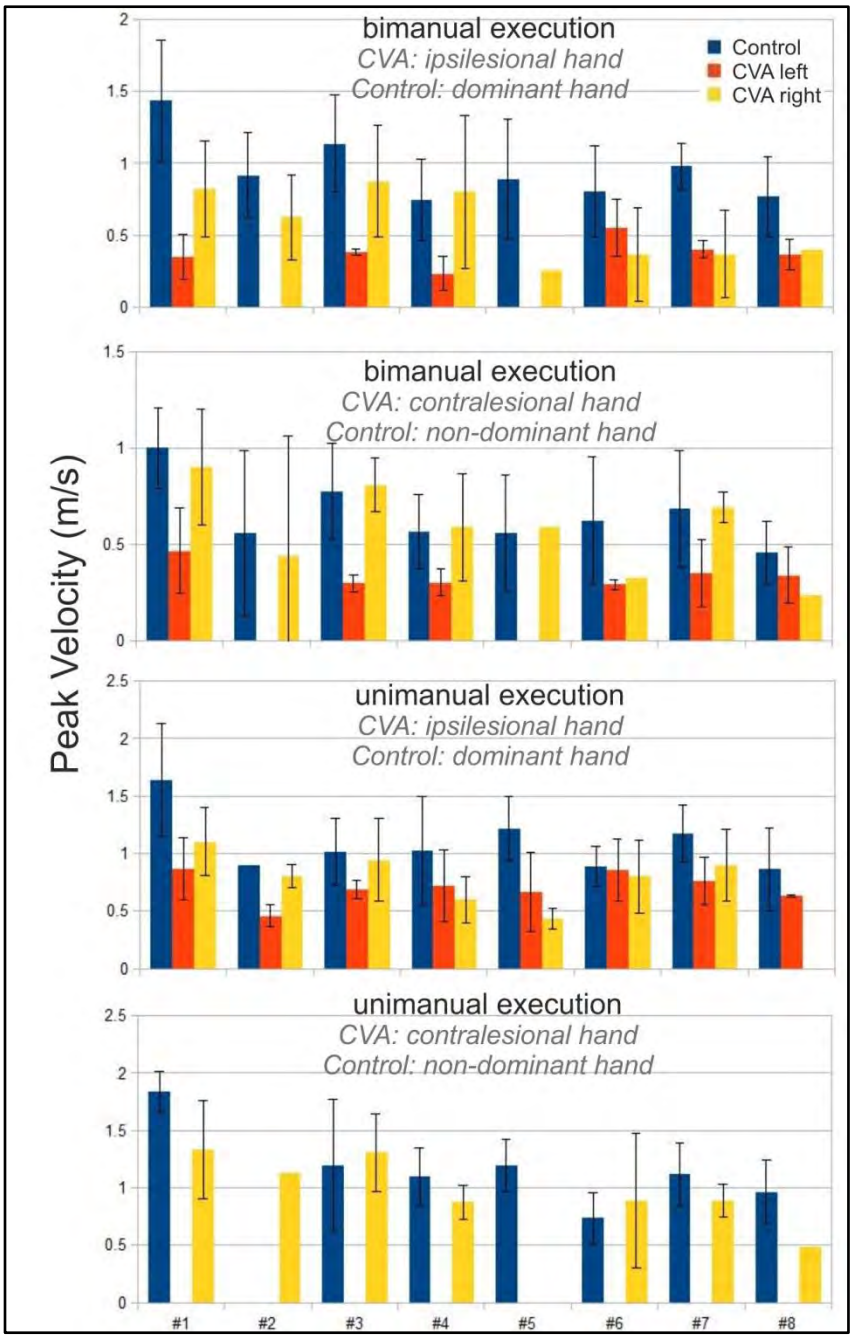


Figure 27: Maximum velocity peaks for the different sub-segments in the three task conditions in controls, LBD & RBD patients.

Patients with left brain damage show extremely low velocity peaks for the different sub-segments in the bimanual dominant / ipsilesional task condition. Patients with right brain damage show lower peaks than the control group but with high standard deviations (Figure 27). In the bimanual task condition, the contralesional hand of RBD patients produce velocity peaks that are overall comparable with the peaks of the control group (Figure 27). LBD patients again yield much lower velocity peaks. Maximum velocity peaks in the unimanual dominant / ipsilesional task condition are higher in the control group than the patient groups for the different sub-segments, but the differences are smaller than in the bimanual condition of the task (Figure 27).In the unimanual non-dominant / contralesional task condition the standard deviations in the RBD patients are large and the velocity peaks are comparable to the controls.

Movement Time versus Path Lengths

Figure 28 shows movement time on the x-axis versus the corresponding path length on the y-axis for the different sub-segments of the bimanual dominant / ipsilesional task condition. The lines indicate the linear regressions for the different groups. For most sub-segments the path length – movement time relation of the groups (representing the average velocity) follows approximately the linear regression which however does not pass through the axes' origin indicating that during shorter paths mean velocity is relatively high. The regressions of the groups differ only slightly in gradients and offset, however, for similar path lengths the duration is significantly prolonged in the patients (e.g. 10 s for longer paths).

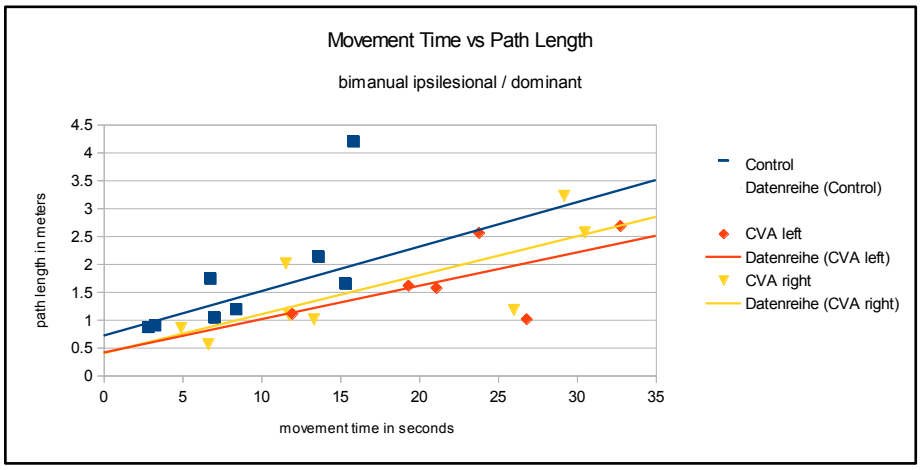


Figure 28: Movement time displayed against path length in the bimanual task condition for the dominant / ipsi-lesional hand in controls, LBD & RBD patients including a linear regression line for each group.

3.4 Discussion

Hand effects in healthy subjects

Path length in the bimanual condition (both hands accumulated) is clearly longer than the path lengths in each unimanual condition, and the movement time is similar or even a bit longer in the bimanual condition compared to the unimanual conditions (Figure 18). Thus the benefit of a bimanual execution is not obvious at first sight. There could be several factors that nevertheless explain the benefits of a bimanual in comparison to a unimanual

execution. One reason could be a better and safer handling of liquids for example. When pouring water, the assistive hand could fixate the target and therefore produce additional path length. Other reasons explaining higher costs in terms of trajectory length and time during bimanual execution could be the avoidance of awkward positions and / or reduction of efforts for the dominant hand. A reduction of effort for the dominant hand by using the non-dominant hand would cause additional path lengths through a shifted starting position (hands return to the body when they are not used). The ratio of the path lengths of dominant and non-dominant hand is quite stable but due to the shifted starting positions, movement time should be raised, which is not the case. The avoidance of awkward positions could be another explanation for the increased path lengths under the bimanual condition. When checking the videos, it is often observed that in the unimanual conditions, subjects use uncommon poses for some of their actions like pouring water into the kettle by using a supination of the right hand. Such awkward positions are avoided in bimanual conditions when the assistive hand adjusts the position of the object for an easier handling at the expense of a longer trajectory.

Kinematic characteristics of sub-segments in healthy subjects

The comparison of kinematics on a sub-segment level shows relatively stable peak velocities, especially of the left hand in bimanual execution (Figure 21). Path length and movement are much more affected by the sub-segment (Figure 17 & Figure 19). Since movement time is the same for the left and right hand, path length is an even more characteristic measure – the path lengths of the left and the right hand are strongly affected by the sub-task. The segments can be coarsely divided into three long (1, 4, 5), three medium (3, 6, 8) and two short segments in execution time (2, 7). Path length show one long (1), three medium (3, 4, 5) and four short (2, 6, 7, 8) segments. Peak velocities are in segment 1 highest and in the other segments relatively constant. Further looking at Figure 25, the linear regression reveals a quite stable increase of movement time with path length (see also Figure 21), but with an offset. This offset seems to have two causes, one is the distribution of the objects on the working surface and the other the size of the objects used.

Segmentation

The segmentation into eight sub-segments seems so far useful, although tea with milk and sugar is quite uncommon for German elderlies. Segment 1 (‘put water in the kettle’) and 2 (‘switch kettle on’) are often not separable from each other in the kinematics but are clearly two sub-tasks as visible in Figure 15. The execution probabilities of segment 1 and 2 are slightly different, even in controls. Segments 5 (‘remove the teabag’) and 8 (‘stir the tea’) are the most interesting segments in terms of omissions. Both patient groups, especially RBD patients, have very low probabilities of executing these sub-segments and also in controls the likelihood to omit is highest in these segments. Interestingly, the overall number of performed sub-segments is lower than eight even in controls (Figure 15). RBD patients show the lowest numbers which is clearly mostly due to segments 5 and 8. The omission of these segments is apparently based on the fact that they are not necessary for succeeding in producing some tea ignoring the quality of the own end-state in the task. The omission of these segments might be a valuable marker for the difficulty of the task for the subject.

Patients' characteristics compared to controls

Overall one can state that the patient groups differ from the control group by slower execution of the task but no increased path length (Figure 22, Figure 24 & Figure 26). The difference between patients with left brain damage and right brain damage is particularly remarkable for the timing and velocities (Figure 26). Movement times are more stable in controls than in the patient groups, but path lengths and peak velocities are similarly variable in all groups. Time differences between the patient groups are rare but meaningful. Patients with right brain damage show shorter movement times for the segments than patients with left brain damage, but similar movement times for the whole task. Remarkably despite they performed less sub-segments (Figure 22, Figure 23 & Figure 28). Apparently RBD patients had longer pauses in between the segments to reconsider their further course of actions. In the sub-segments they show maximum velocity peaks not much lower than the controls (Figure 26). Patients with left brain damage on the other side have long movement times in total, comparable to patients with right brain damage, but path lengths similar to the control group (Figure 22 & Figure 24). Their movement speed however is very low, visible in the low maximum peak velocities in particularly in the bimanual condition and also for the ispilesional hand (Figure 26). Concluding patients with right brain damage make more pauses and patients with left brain damage move slower than controls.

Hand effects in patients

As the control subjects, patients show shorter overall movement times in their unimanual task conditions. Especially RBD patients have their shortest time to perform the task in the contralesional condition (Figure 22). This fast task execution could come from intense training with their impaired hand with nearly 4years after stroke. The overall movement time in the patient groups is comparable in the bimanual and unimanual ipsilesional task condition. Looking at the path lengths, patients show the same characteristics as controls in path lengths of bimanual and unimanual trials (Figure 24). A remarkable feature of the LDB patients is that according to the path lengths of right and left hand in the bimanual condition they seem to act as slight right handers. The path length of their right hand, although contralesional and typically paretic, is longer than the path length of their left, ipsilesional, hand. This is only switched in segments 3 ('place a teabag in the mug') and 6 ('add milk') (Figure 25). Again this could be at least partially explainable through an additional training of the contralesional hand, but still two segments are not in accordance to that phenomenon. For the milk an explanation could be that the jug is placed left to the patient on the working surface and reaching for it with the right hand would result in awkward positions. Since the teabags are placed in front of the subjects, there must be another basis for this behavior. Teabags can behave like a pendulum when grasped by the label. Accordingly, their controlled placement demands fine motor performance and the patients use their ipsilesional hand to succeed in this sub-task.

Maximum velocity peaks do strongly differ for LBD patients between bimanual and unimanual ipsilesional task conditions (Figure 26). In the unimanual ipsilesional condition they are restricted to only use their functional hand and therefore reach higher speeds than in the bimanual condition where they try to use their impaired - and formerly dominant - hand as much as possible. Obviously higher speed of the ipsilesional hand is successfully enforced by the condition.

Patients' performance in the sub-segments

Sub-segment characteristics of the patients are quite comparable to the controls' in terms of path length. Figure 28 shows that patients have similar path lengths but with longer movement times for the sub-segments displayed as a shift to the right. Also they exhibit more variance along the x-axis. So the average velocities in the sub-segments are depending on the sub-segment but follow a rule that is based on the path demands of the sub-segment.

On the sub-segment level RBD patients show prolonged movement times in comparison to LBD patients in segments 4 ('pour heated water into the mug') and 7 ('add one sugar cube'), while LBD patients have prolonged movement times for segment 3 ('place a teabag in the mug'), 5 ('remove the teabag') and 6 ('add milk') (Figure 23). Interestingly segments 4 and 7 use objects that are right from the subject's midline and RBD patients should be fine with their ipsilesional hand. Segments 3, 5 and 6 succeed either on the left side of the subject or include using the teabags and LBD patients are here using their ipsilesional, unimpaired hand. A prolonged movement time in this case is hard to explain but a possible solution could be active inhibition to not using their contralesional hand. Path lengths of the sub-segments only show anomalies in the bimanual condition for the ipsilesional hand. Here RBD patients have longer path lengths in segment 4 ('pour heated water into the mug') and LBD patients in segment 6 ('add milk') and both groups lower path lengths than controls in segment 1 ('put water into the kettle') (Figure 25). Despite all three segments include pouring movements and transportations of containers filled with a liquid they have very different path lengths. It may be that patients produce a straighter and slower trajectory in segment 1, while controls that are moving the container with higher speed and have to describe a more curved movement path. Segment 6 in LBD patients and segment 4 in RBD patients are both executed with their ipsilesional, functional hand and therefore they are using a more similar trajectory to the controls, although they are still slowed.

Conclusions for action recognition

The results for the patient group indicate that an action recognition algorithm trained with data from healthy subjects would also work on stroke patients if the system is not sensitive to prolonged execution times and decreased velocity peaks. Technical problems with latencies of action recognition would even be reduced in patients with longer movement times and lower peak velocities. Although path length and movement time are very variable in all groups, their relation follows a more or less constant proportion (Figure 28). The path lengths and therefore the trajectory of the patients' movements are the most comparable measure to the control subjects and this seems to offer a promising opportunity for the action recognition system. This knowledge can potentially improve approaches to action recognition using marker-based or video based methods. Additional data collected from object coasters and modelling (HMMs) could be used to increase the reliability of the action recognition system allowing scope for implementation as a home-based rehabilitation system. For implementing further ADLs training data of healthy subjects could be sufficient, so that the capability of the system can easily be expanded.

4. ACTION RECOGNITION FOR TOOTH BRUSHING

4.1 Acoustic identification of tooth-brush position location

4.1.1 The second CogWatch prototype system

The second CogWatch prototype system is a rehabilitation system designed to re-train stroke patients to brush their teeth. One of the main challenges in this application is to automatically identify the location of the head of the toothbrush in the mouth, so that the system can monitor that the teeth are brushed in all locations in the mouth.

4.1.2 Categorization of mouth positions

A pilot study was conducted into automatic identification of the position of the toothbrush head in the mouth. For the purposes of this study it was necessary to partition the mouth cavity into a number of distinct regions, which define the 'classes' for automatic classification.

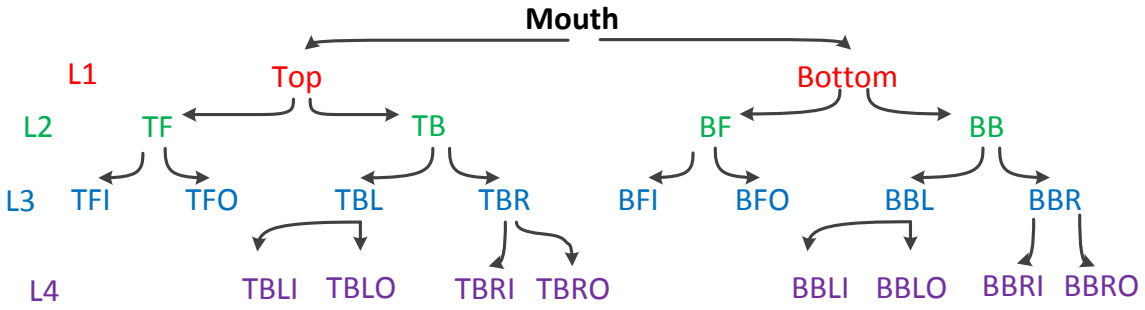


Figure 29: Hierarchical description of the partition of the mouth cavity into different location for tooth-brushing.

Figure 29 shows a hierarchical tree representation of the partition of the mouth into regions for toothbrush head position location. Level 1 is split into Top and Bottom. Level 2 adds Front and Back. Level 3 splits classes at the front into Inner & Outer and classes at the back into Left & Right. Level 4 splits all back classes into Inner & Outer. A particular position is labelled using an acronym that indicates the path through the hierarchy which is required to reach that position. For instance, BBRO stands for Bottom-Back-Right-Inner.

4.1.3 Automatic identification of toothbrush head location

A number of approaches to this problem are being evaluated, including the use of an instrumented toothbrush that has an accelerometer, gyroscope and magnetometer embedded in its handle, and the use of hand tracking systems based on Kinect or Leap.

An alternative and novel approach is to use audio recordings of tooth-brushing. The audio data would be captured using an external microphone, either mounted externally in front of the subject's mouth (for example, a small microphone could be embedded in a bathroom mirror) or head-mounted. It might also be possible to use an array of microphones with beam-forming to achieve some degree of noise robustness.

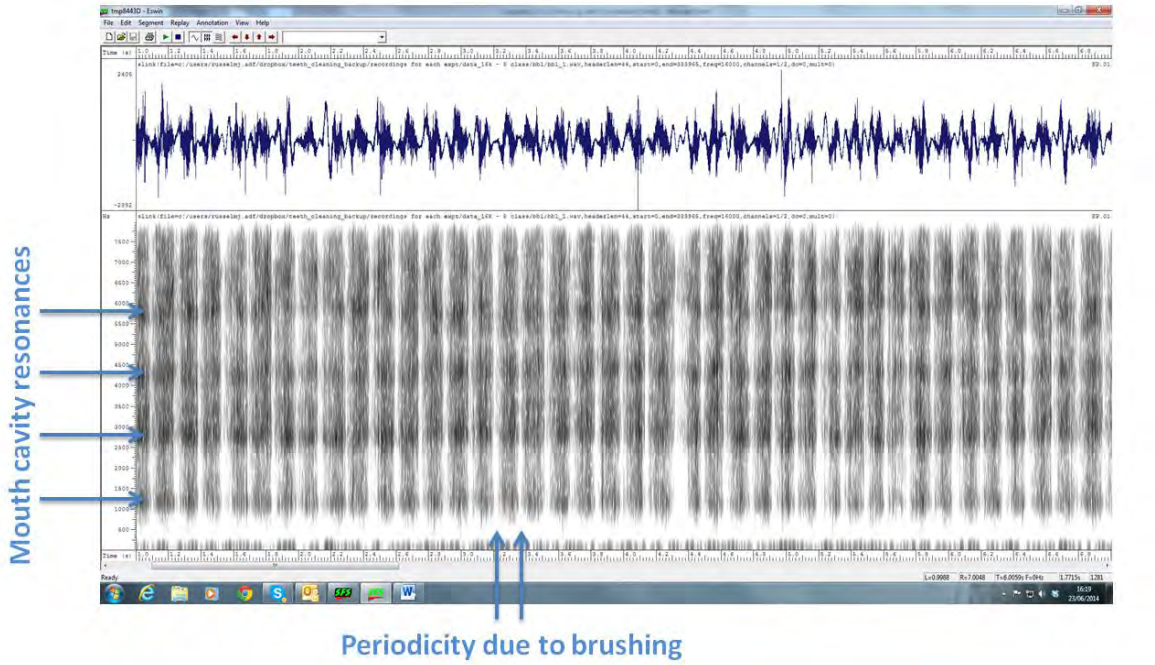


Figure 30: Spectrogram of a five second recording of brushing teeth in the back-bottom-left region of the mouth.

There are two motivations for this approach. First, even if the mouth shape is kept constant, the audio signal will depend to some extent on the location of the toothbrush head in the mouth. Second, in reality the mouth shape will not be kept constant and instead the jaw and tongue will move during tooth-brushing to facilitate access to different teeth. This will change the resonance properties of the mouth cavity, which will change the properties of the acoustic signal. More specifically we can think of the tooth-brush head as a sound source during brushing, and the resonances of the mouth cavity will modify the spectrum of that sound in different ways as the mouth shape changes.

4.2 A corpus of recordings of tooth-brushing

4.2.1 Audio data collection

A corpus of audio recordings of tooth-brushing was collected. Each recording corresponded to brushing in a particular location in the mouth in Figure 29. The recordings are stereo and sampled at 16kHz. The left channel was linked to a head-mounted microphone, with the microphone cell positioned about 2cm from the mouth on the right side. The right channel was linked to the remote omnidirectional microphone, positioned about 30cm directly in front of the mouth.

The recordings are catalogued in Table 3. They consist of 80 recordings, each of approximately 15 – 20 seconds duration, totaling 1440.8 seconds. The recordings were all made by the same individual using the same toothbrush.

4.2.2 Example acoustic signals for tooth brushing

Figure 30 shows a spectrogram of a five second recording on an individual brushing teeth in the back, bottom left-hand side of his mouth. For those who are not familiar with a spectrogram, the horizontal axis is time (in seconds) and the vertical axis is frequency in Hz. The recording was sampled at 16kHz, and so the maximum frequency component is at 8kHz. The grey scale represents the power at a particular time and frequency. The spectrogram was created using the UCL Speech Filing System (SFS)[1].

Two elements of structure are evident from the figure. The regular vertical bands correspond to the dynamics of brushing, with the gaps between the bands indicating times when the toothbrush was at rest. The fact that there are approximately 7 of these bands per second indicates a brushing frequency of about 3.5Hz. The less evident horizontal bands indicate prominent frequencies in the audio signal. Our premise is that at least some of these are due to the resonances of the mouth cavity and hence will change as the shape of the mouth cavity changes.

For back-bottom-left brushing (Figure 30), these resonances occur at approximately 1200 Hz, 2800 Hz, 4250 Hz and 6750 Hz.

Class	Number of Recordings	Average Recording Duration (sec)	Total Duration of Recordings (sec)
BBL	12	16.7	201.0
BBR	12	16.5	198.5
BFI	8	18.5	147.7
BFO	8	21.1	168.9
TBL	12	18.0	216.5
TBR	12	18.6	222.6
TFI	8	16.0	128.0
TFO	8	19.7	157.6

Table 3: Statistics of the recordings of tooth-brushing that were made for the pilot experiment.

From the perspective of speech production, the broad band of high frequency energy in Figure 31 is similar to that which one would expect to be present in a sound such as /s/, which is articulated by creating turbulence at the front of the mouth with the teeth (almost)

[11] www.phon.ucl.ac.uk/resource/sfs/

closed. Figure 30 is more similar to the spectrogram that one would expect to observe for a vowel sound.

Figure 31 shows the corresponding spectrogram for a five second recording of brushing in the top-front-outside of the mouth. In this case there are resonances at approximately 1250 Hz, 3200 Hz and 4750 Hz. The highest frequency resonance in Figure 30 is not evident in Figure 31. However, Figure 31 shows a broad range of high energy over high frequencies. Because the brushing in Figure 31 is at the front of the mouth, and because the brushing is on the outside of the teeth, the mouth may be closed and in this cased one would expect the resonance frequencies of the mouth to exert less influence.

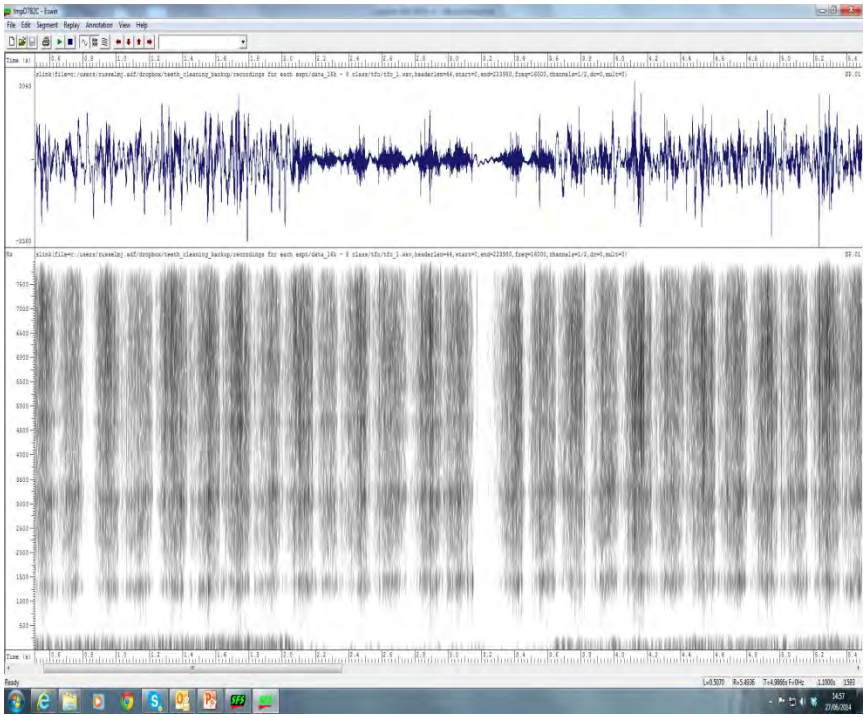


Figure 31: Spectrogram of a five second recording of brushing teeth in the top-front-outside region of the mouth

Figure 32 shows a spectrogram for a five second recording on teeth-cleaning with the brush cleaning the top-front-inside of the mouth. Because the mouth is at least partially open, the resonant frequencies in the high frequency regions are more evident than in Figure 31 (though they are faint).

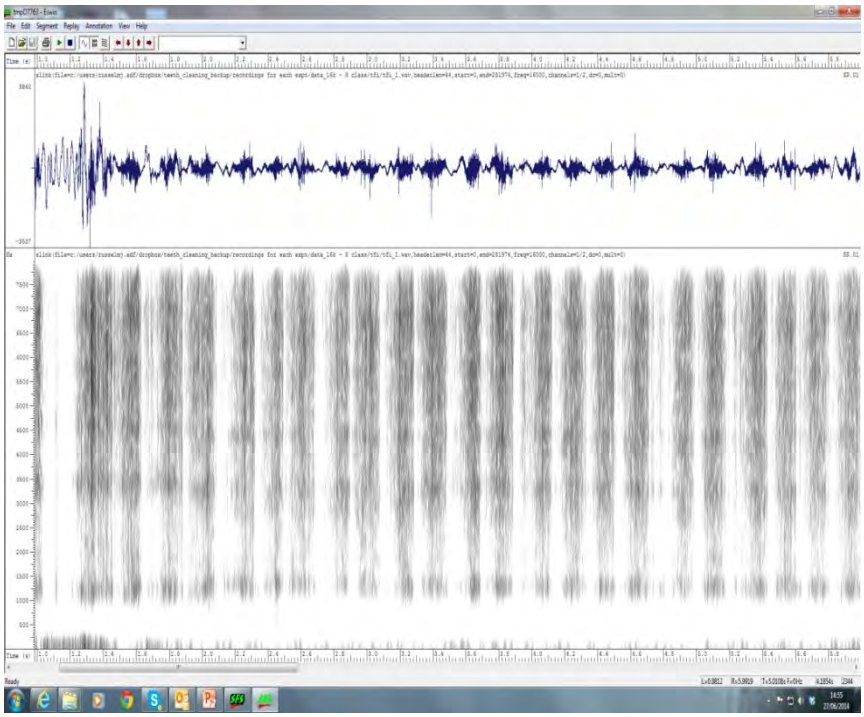


Figure 32: Spectrogram of a five second recording of brushing teeth in the top-front-inside region of the mouth

4.2.3 Initial conclusions

This analysis appears to support the hypotheses that the different mouth positions adopted for brushing teeth in different locations give rise to distinct resonance properties, and that these distinctions are sufficient to distinguish between the different brushing locations from the audio signal alone.

The next section presents the results of automatic classification experiments to test this hypothesis.

4.3 Automatic identification of tooth-brushing position from audio data

4.3.1 Method

This section describes a set of experiments in automatic tooth-brush head position detection using the data described in Section 4.2. The experiment uses a simple statistical pattern recognition system based on Gaussian Mixture Models (GMMs).

4.3.1.1 Feature extraction

The first step is to apply feature extraction to each of the audio recordings. The objective is to convert the recording into a sequence of feature vectors, where the type of feature vector

is chosen to emphasize properties of the acoustic signal that are useful for classification and to suppress those that are not. The number of potential analyses is huge and at this stage it is not possible to evaluate all of them experimentally. Hence the particular method that is adopted is the most common approach to feature extraction used in automatic speech recognition.

These feature vectors, called Mel Frequency Cepstral Coefficients (MFCCs) are based on a spectral analysis of the signal, and would therefore be expected to exhibit the types of differences in resonance structure described in Section 4.2.2. The procedure for calculating MFCC vectors is described fully in the HTK Book (Young et al. 2006) and many other sources.

Briefly, the first 25 ms of recording are considered and a Hamming window is applied, followed by a discrete Fourier transform (DFT). The modulus and then the logarithm of the DFT values are taken (so that phase information is ignored). The frequency scale of the DFT spectrum is converted into a non-linear perceptually-motivated scale called the mel scale. For a 16 kHz signal, this typically results in a 26 point mel frequency log DFT. Finally, a discrete cosine transform is applied to give a 26 point cepstrum, the high order cepstral coefficients are discarded, and the remaining 'static' cepstral features are supplemented with approximations to their first and second order derivatives. The resulting vector typically has 39 dimensions. The window is then shifted along in time by 10ms, and the whole process is repeated. This results in a sequence of 39 dimensional mel frequency cepstral vectors (MFCC vectors), one every 10ms. For more details please refer to the HTK book (Young et al. 2006).

4.3.1.2 Statistical modelling

For each tooth-brushing location of interest, a multivariate probability density function (PDF) is constructed which characterizes the distribution of MFCC vectors for recordings of tooth-brushing in that location. In principle one could use a multivariate Gaussian PDF for this purpose. However, experience from speech recognition suggests that the true distribution of data is not unimodal, and hence that a single multivariate Gaussian PDF may be too simple. The standard solution is to use a multivariate Gaussian mixture PDF, or Gaussian Mixture Model (GMM).

An M component GMM is a PDF p of the form:

$p(y) = \sum_{m=1}^{M} w_m g_m(y)$,

where y is a feature vector, $w_1, \dots, w_M$ are real numbers between 0 and 1 satisfying

$\sum_{m=1}^{M} w_m = 1$,

and each g_m is a multivariate Gaussian PDF.

The parameters of an M component GMM are the means $\mu_1, \dots, \mu_M$ and covariance matrices $\sigma_1, \dots, \sigma_M$ of its component PDFs and the mixture weights $w_1, \dots, w_M$. These parameters are estimated automatically from data using the E-M algorithm.

As training data is typically limited it is often assumed that the covariance matrices σ_m are diagonal, and this assumption was made in these experiments.

4.3.2 Experiments

4.3.2.1 Experiment 1: Two-classes, front vs back

In this experiment a total of 72 recordings were used, totaling 1074.7 seconds in duration. The recording statistics are shown in Table 4. All recordings were taken from those listed in Table 3.

Class	Number of Recordings	Average Recording Duration (sec)	Total Duration of Recordings (sec)
Front	30	15.2	455.8
Back	42	14.7	618.9

Table 4: Recordings used in experiment 1.

This experiment was composed of 30 unique sub-experiments from which an average accuracy was determined. In each sub-experiment, the models were trained with 29 Front recordings and 41 Back recordings. The models were tested with 1 Front recording and 1 Back Recording in each sub-experiment. In this way the amount of data available for training was maximized and the test data was never included in the training data.

The experiment was repeated using models with different numbers of GMM components. The results are shown in Figure 33.

As one would expect, the results on the training data (blue graph) are better than those for the test data (green graph). The performance on the test and training sets improves as the number of GMM mixture components is increased up to 44, beyond which the classification rate on the test set is close to 100% (the figure of 98.33% for a 64 component GMM corresponds to 1 error).

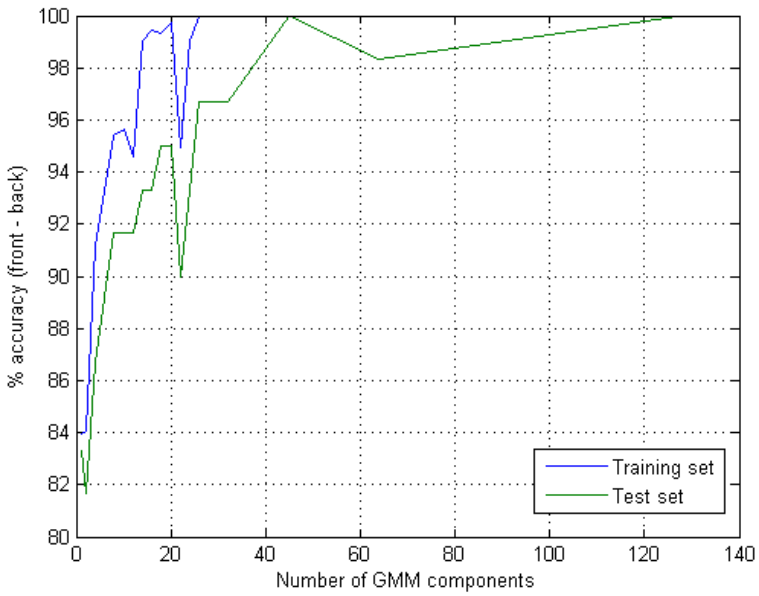


Figure 33: Results of classification experiments to distinguish between tooth-brushing at the front and back of the mouth.

4.3.2.2 Experiment 2: Two-classes, top vs bottom

In this experiment a total of 104 recordings were used, totaling 1904.8 seconds in duration. The recording statistics are shown in Table 5. All recordings were taken from those listed in Table 3.

Class	Number of Recordings	Average Recording Duration (sec)	Total Duration of Recordings (sec)
Top	52	18.8	978.0
Bottom	52	17.8	926.8

Table 5: Recordings used in experiment 2.

This experiment was composed of 52 unique sub-experiments from which an average accuracy was determined. In each sub-experiment, the models were trained with 51 Top recordings and 51 Bottom recordings. The models were tested with 1 Top recording and 1 Bottom recording in each sub-experiment. The test data was completely new to the system.

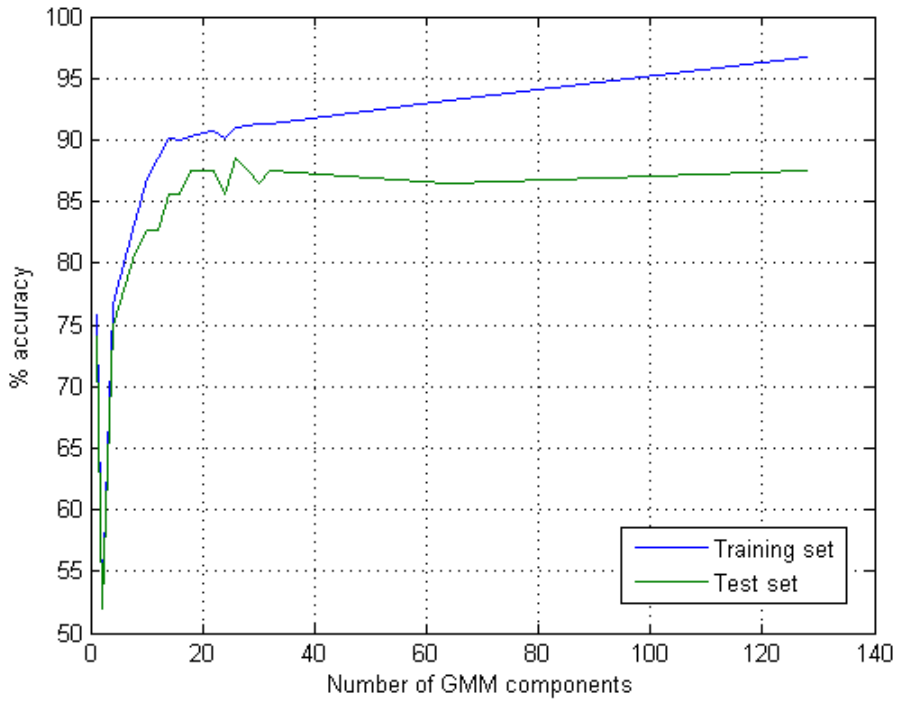


Figure 34: Results of classification experiments to distinguish between tooth-brushing at the top and bottom of the mouth.

The results are shown in Figure 34. The classification accuracies for 64 and 128 component GMMs are 86.54% and 87.5% respectively.

4.3.2.3 Experiment 3: Three classes, back-right, back-left and front

In this experiment a total of 100 recordings were used, totaling 1877 seconds in duration (Table 6).

This experiment was composed of 30 unique sub-experiments from which an average accuracy was determined. In each sub-experiment, the models were trained with 29 BL recordings, 29 BR recordings and 39 Front recordings. The models were tested with 1 BL recording, 1 BR Recording and 1 Front recording – in each sub-experiment. The test data was completely new to the system.

The experiment was repeated using models with different numbers of GMM components. The results are shown in Figure 35.

Class	Number of Recordings	Average Recording Duration (sec)	Total Duration of Recordings (sec)
BL	30	17.9	536.6
BR	30	18.3	550.5
Front	40	19.7	790.0

Table 6: Recordings used in experiment 3.

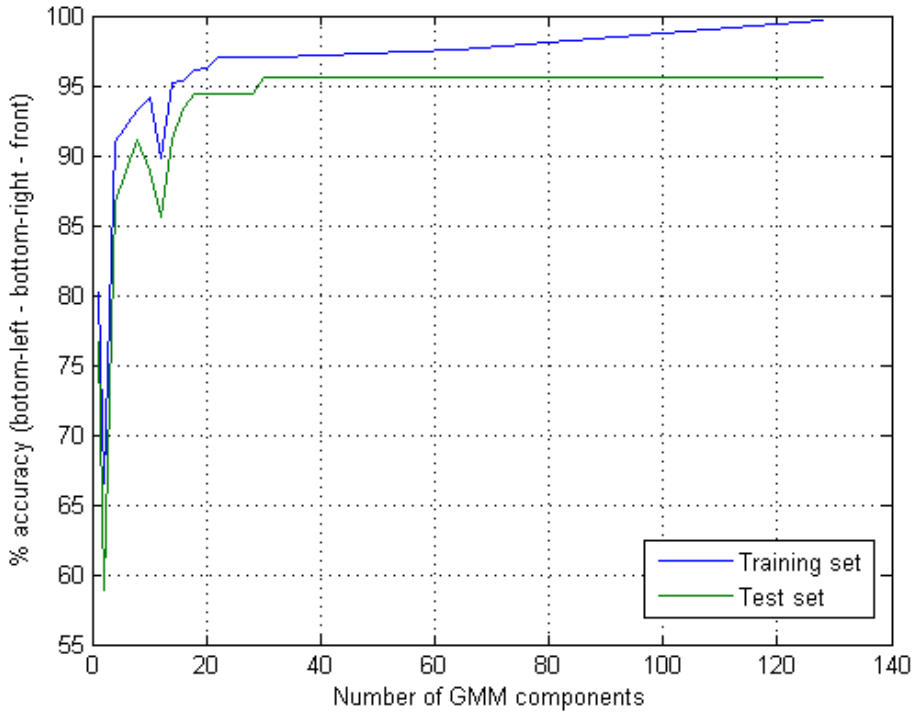


Figure 35: Results of classification experiments to distinguish between tooth-brushing at the back-right, back-left and front of the mouth.

The accuracy on the test set is 95.6% for GMMs with 32 or more components.

4.3.2.4 Experiment 4: Four classes, back-right, back-left, front-inside and front-outside.

In this experiment a total of 92 recordings were used, totaling 1689.3 seconds in duration. The data is listed in Table 7.

Class	Number of Recordings	Average Recording Duration (sec)	Total Duration of Recordings (sec)
BL	30	17.9	536.6
BR	30	18.3	550.5
Front-Inside	16	17.2	275.7
Front-Outside	16	20.4	326.5

Table 7: Recordings used in experiment 4.

This experiment was composed of 16 unique sub-experiments from which an average accuracy was determined. In each sub-experiment, the models were trained with 29 BL recordings, 29 BR recordings, 15 Front-In recordings and 15 Front-out recordings. The models were tested with 1 BL recording, 1 BR Recording, 1 Front-In recording and 1 Front-Out recording – in each sub-experiment. The test data was completely new to the system.

The results are shown in Figure 36. For 64 and 128 component GMMs, the classification accuracy is 92.2%.

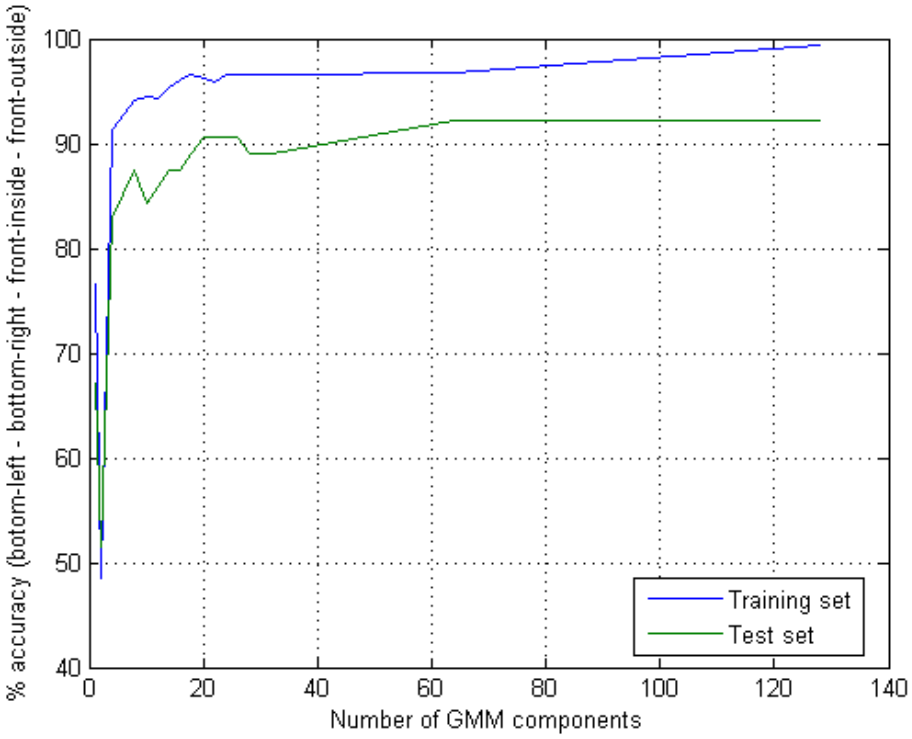


Figure 36: Results of classification experiments to distinguish between tooth-brushing at the back-right, back-left, front-inside and front-outside of the mouth.

4.3.2.5 Experiment 5: Six classes – front-inside, front-outside, bottom-back-left, bottom-back-right, top-back-left and top-back-right.

In this experiment a total of 80 recordings were used, totaling 1440.8 seconds in duration (Table 8).

Class	Number of Recordings	Average Recording Duration (sec)	Total Duration of Recordings (sec)
Front-Inside	16	17.2	275.5
Front-Outside	16	20.4	326.5
BBL	12	16.7	201.0
BBR	12	16.5	198.5
TBL	12	18.0	216.5
TBR	12	18.6	222.6

Table 8: Recordings used in experiment 5.

This experiment was composed of 12 unique sub-experiments from which an average accuracy was determined. In each sub-experiment, the models were trained with 15 Front-In recordings, 15 Front-Out recordings, 11 BBL recordings, 11 BBR recordings and 11 TBR recordings. The models were tested with 1 recording of each class – in every sub-experiment. The test data was completely new to the system.

The experiment was repeated using models with different numbers of GMM components. The results are shown in Figure 37. The classification accuracies on the test set for 64 and 128 component GMMs are 84.7% and 86.1%, respectively.

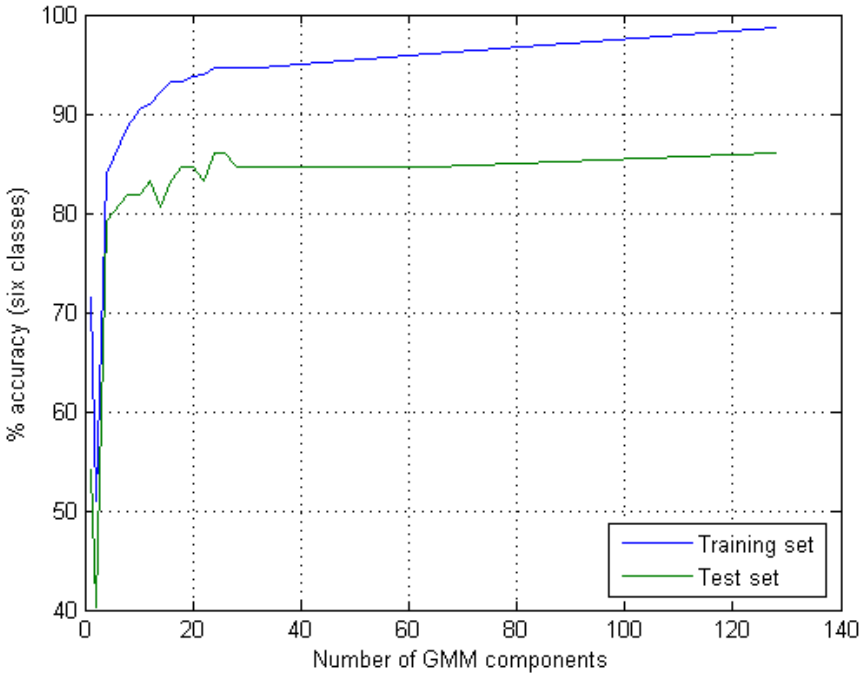


Figure 37: Results of classification experiments to distinguish between tooth-brushing at six different locations: the bottom-back-right, bottom-back-left, top-back-right, top-back-left, front-inside and front-outside of the mouth.

4.3.2.6 Experiment 6: Eight classes – bottom-front-inside, bottom-front-outside, top-front-inside, top-front-outside, bottom-back-left, bottom-back-right, top-back-left and top-back-right.

Class	Number of Recordings	Average Recording Duration (sec)	Total Duration of Recordings (sec)
BBL	12	16.7	201.0
BBR	12	16.5	198.5
BFI	8	18.5	147.7
BFO	8	21.1	168.9
TBL	12	18.0	216.5
TBR	12	18.6	222.6
TFI	8	16.0	128.0
TFO	8	19.7	157.6

Table 9: Recordings used in experiment 6.

The results of experiment 6 are shown in Figure 38. The figure shows recognition accuracies on the test data of 82.8% and 84.4% for GMMs with 64 and 128 components, respectively.

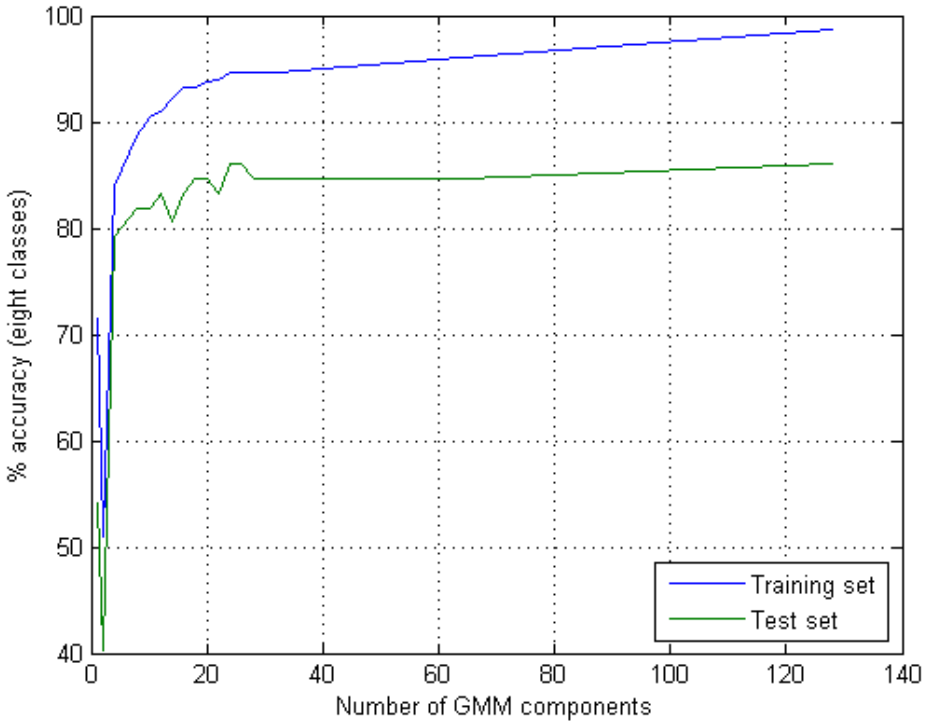


Figure 38: Results of classification experiments to distinguish between tooth-brushing at eight different locations: the bottom-front-inside, bottom-front-outside, top-front-inside, top-front-outside, bottom-back-left, bottom-back-right, top-back-left and top-back-right.

4.3.3 Summary of classification experiment results

Table 10 shows a summary of the results from experiments 1 to 6. The table shows results on the test set (columns 4 and 5) and training set (column 6 and 7) that were obtained using 64 and 128 component GMMs.

Expt.	Num Class	Classes	64 Comp (Test)	128 Comp (Test)	64 Comp (Train)	128 Comp (Train)
1	2	Front / back	98.3	100.0	100.0	100.0
2	2	Top / bottom	86.5	87.5	93.2	96.7
3	3	Back-right / back-left / front	95.6	95.6	97.5	99.7
4	4	Back-right / back-left / front-in / front-out	92.2	92.2	96.7	99.4
5	6	Bottom-back-right / bottom-back-left / top-back-right / top-back-left / front-in / front-out	84.7	86.1	96.1	98.8
6	8	Bottom-back-right / bottom-back-left / top-back-right / top-back-left / bottom-front-in / bottom-front-out / top-front-in / top-front-out	82.8	84.4	96.2	98.8

Table 10: Summary of classification results.

The discussion focusses on the results for the test sets.

Overall, the results are much better than anticipated. However, it must be remembered that these experiments are restricted to recordings from a single individual using the same toothbrush in a quiet environment.

It is evident from the table (experiments 1 and 2) that differentiating between brushing in the top and bottom areas of the mouth is more difficult that differentiating between the front and back regions. This suggests that the differences between the mouth cavity shapes adopted for front and back brushing are greater than the differences for top and bottom brushing, at least in terms of their effects on the resonances of the mouth cavity, and this is consistent with intuition. However, the difficulty seems to be mainly in differentiating between top and bottom at the back of the mouth. Adding the top / bottom distinction at the front of the mouth (which is the difference between experiment 5 and experiment 6) results in only a relatively modest increase in error rate (from 15.3% to 17.2%).

The difficulty of distinguishing between corresponding top and bottom locations in the mouth also explains the higher error rates in experiment 5 compared with experiment 4 (experiment 5 the same as experiment 4, except that it also involves top / bottom distinctions for brushing at the back of the mouth). In this case adding the top / bottom distinction approximately doubles the error rate from 7.8% (experiment 4) to 15.3%.

Comparing experiments 1 and 3 shows the effect of distinguishing between the left- and right-sides of the back of the mouth (“front / back” versus “front / back-right / back-left”). Focusing on the system with 64 components, adding this distinction results in a 150% increase in error-rate, from 1.7% to 4.4%.

Experiment 4 requires differentiation between inside and outside brushing at the front of the mouth, compared with experiment 3.

In conclusion, acoustic-based identification of tooth-brush head position in the mouth appears to be a promising approach. “Top – bottom” distinctions appear to be the most

difficult, with an error rate of 15.5% compared with 1.7% for “front – back”. “Left-right” distinctions are also relatively easy.

Of course, in a real application the system would need to be able to deal with different users, different tooth-brushes and environmental noise. However, it must be remembered that in a real application acoustic-based recognition would not be used on its own but in combination with other classifiers. The key question is how similar are the errors made by acoustic-based classification and more conventional sensor-based classification, and whether their outputs can be successfully fused to give improved results.

5. REFERENCES

Humphreys, G. W. & Forde, E. M. E. (1998). Disordered action schema and action disorganisation symdrome. In Forde, E. M. E., Rusted, J., Mennie, N., Land, M. & Humphreys, G. W. (2010). The eyes have it: An exploration of eye movements in action disorganisation syndrome. Neuropsychologia 48, 1895-1900.

Hughes, C. M., Parekh, M., & Hermsdörfer, J. (2013). *Segmenting instrumented activities of daily living (IADL) using kinematic and sensor technology for the assessment of limb apraxia.* Paper presented at the HCI International 2013-Posters' Extended Abstracts, Las Vegas.

Young, S.J., Evermann, G., Gales, M.J.F., Hain, T., Kershaw, D., Moore, G., Odell, J., Ollason, D., Povey, D., Valtchev, V., Woodland, P.C., (2006) *The HTK Book, version 3.4,* Cambridge University Engineering Department.