PROCEEDINGS

1ST WORKSHOP ON

MACHINE LEARNING FOR INTERACTIVE SYSTEMS: BRIDGING THE GAP BETWEEN LANGUAGE, MOTOR CONTROL AND VISION

(MLIS-2012)

Heriberto Cuayáhuitl, Lutz Frommberger, Nina Dethlefs, Hichem Sahli (eds.)

August 27, 2012

20th European Conference on Artificial Intelligence (ECAI)

Montpellier, France

1ST WORKSHOP ON MACHINE LEARNING FOR INTERACTIVE SYSTEMS (MLIS-2012): BRIDGING THE GAP BETWEEN LANGUAGE, MOTOR CONTROL AND VISION

ORGANIZERS

Heriberto Cuayáhuitl
German Research Center for Artificial Intelligence (DFKI), Saarbrücken, Germany
hecu01@dfki.de

Lutz Frommberger
Cognitive Systems Research Group, University of Bremen, Germany
lutz@informatik.uni-bremen.de

Nina Dethlefs
Heriot-Watt University, Edinburgh, UK
n.s.dethlefs@hw.ac.uk

Hichem Sahli
Vrije Universiteit Brussel, Belgium
hichem.sahli@vub.de

PROGRAM COMMITTEE

Maren Bennewitz, University of Freiburg, Germany
Martin Butz, University of Tübingen, Germany
Paul Crook, Heriot-Watt University, Edinburgh, UK
Mary Ellen Foster, Heriot-Watt University, Edinburgh, UK
Konstantina Garoufi, University of Potsdam, Germany
Milica Gašić, Cambridge University, UK
Helen Hastie, Heriot-Watt University, Edinburgh, UK
Jesse Hoey, University of Waterloo, Canada
Srinivasan Janarthanam, Heriot-Watt University, Edinburgh, UK
Filip Jurčíček, Charles University in Prague, Czeck Republic
Simon Keizer, Heriot-Watt University, Edinburgh, UK
Shanker Keshavdas, German Research Centre for Artificial Intelligence, Germany
Kazunori Komatani, Nagoya University, Japan
George Konidaris, Massachusetts Institute of Technology, USA
Ivana Kruijff-Korbayová, German Research Centre for Artificial Intelligence, Germany
Ramon Lopez de Mantaras, Spanish Council for Scientific Research, Spain
Pierre Lison, University of Oslo, Norway
Iván V. Meza, National Autonomous University of Mexico, Mexico
Roger Moore, University of Sheffield, UK
Eduardo Morales, National Institute of Astrophysics, Optics and Electronics, Mexico
Justus Piater, University of Innsbruck, Austria
Olivier Pietquin, Supélec, France
Matthew Purver, Mary Queen University London, UK
Antoine Raux, Honda Research Institute, USA
Verena Rieser, Heriot-Watt University, Edinburgh, UK

Raquel Ros, Imperial College London, UK
Alex Rudnicky, Carnegie Mellon University, USA
Hiroshi Shimodaira, University of Edinburgh, UK
Danijel Skočaj, University of Ljubljana, Slovenia
Enrique Sucar, National Institute of Astrophysics, Optics and Electronics, Mexico
Martijn van Otterlo, Radboud University Nijmegen, Netherlands
Jason Williams, Microsoft Research, USA
Junichi Yamagishi, University of Edinburgh, UK
Hendrik Zender, German Research Centre for Artificial Intelligence, Germany

Organizing Institutions

Language Technology Lab
German Research Center for Artificial Intelligence (DFKI), Saarbrücken, Germany
http://www.dfki.de

Cognitive Systems Research Group
University of Bremen, Germany
http://www.sfbtr8.spatial-cognition.de

Interaction Lab
Heriot-Watt University, Edinburgh, UK
http://www.hw.ac.uk

Department of Electronics and Informatics
Vrije Universiteit Brussel, Belgium
http://www.vub.be

Sponsors

SFB TR8 Spatial Cognition
http://www.sfbtr8.uni-bremen.de

EUCOG (European Network for the Advancement of Artificial Cognitive Systems, Interaction and Robotics)
http://www.eucognition.org

Additional Sponsors

European FP7 Project - ALIZ-E ICT-248116
http://www.aliz-e.org/

European FP7 Project - PARLANCE 287615
https://sites.google.com/site/parlanceprojectofficial/

TABLE OF CONTENTS

SCHEDULE

Monday, August 27, 2012

09:15 - 09:30 Welcome and opening remarks

09:30 - 10:30 Invited Talk: **Oliver Lemon**
Data-driven Methods for Adaptive Multimodal Interaction

10:30 - 11:00 Coffee break

Session 1: INTERACTIVE SYSTEMS

11:00 - 11:30 *Learning Hierarchical Prototypes of Motion Time Series for Interactive Systems*
Ulf Großekathöfer, Shlomo Geva, Thomas Hermann and Stefan Kopp

11:30 - 12:00 *A Corpus Based Dialogue Model for Grounding in Situated Dialogue*
Niels Schütte, John Kelleher and Brian Mac Namee

12:00 - 12:30 *Towards Optimising Modality Allocation for Multimodal Output Generation in Incremental Dialogue*
Nina Dethlefs, Verena Rieser, Helen Hastie and Oliver Lemon

12:30 - 14:00 Lunch break

14:00 - 15:00 Invited Talk: **Jeremy Wyatt**
Autonomous Learning in Interactive Robots

Session 2: INTERACTIVE ROBOTS

15:00 - 15:20 *Machine Learning of Social States and Skills for Multi-Party Human-Robot Interaction*
Mary Ellen Foster, Simon Keizer, Zhuoran Wang and Oliver Lemon

15:20 - 15:40 *Hierarchical Multiagent Reinforcement Learning for Coordinating Verbal and Nonverbal Actions in Robots*
Heriberto Cuayáhuitl and Nina Dethlefs

15:40 - 16:10 Coffee break

16:10 - 16:30 *Using Ontology-based Experiences for Supporting Robots Tasks - Position Paper*
Lothar Hotz, Bernd Neumann, Stephanie Von Riegen and Nina Worch

16:30 - 16:50 *Fast Learning-based Gesture Recognition for Child-Robot Interactions*
Weiyi Wang, Valentin Enescu and Hichem Sahli

16:50 - 17:30 Panel discussion and closing remarks

PREFACE

Intelligent interactive agents that are able to communicate with the world through more than one channel of communication face a number of research questions, for example: how to coordinate them in an effective manner? This is especially important given that perception, action and interaction can often be seen as mutually related disciplines that affect each other.

We believe that machine learning plays and will keep playing an important role in interactive systems. Machine Learning provides an attractive and comprehensive set of computer algorithms for making interactive systems more adaptive to users and the environment and has been a central part of research in the disciplines of interaction, motor control and computer vision in recent years.

This workshop aims to bring researchers together that have an interest in more than one of these disciplines and who have explored frameworks which can offer a more unified perspective on the capabilities of sensing, acting and interacting in intelligent systems and robots.

The MLIS-2012 workshop contains papers with a strong relationship to interactive systems and robots in the following topics (in no particular order):

- sequential decision making using (partially observable) Markov decision processes;
- multimodal dialogue optimization and information presentation using flat or hierarchical (multiagent) reinforcement learning;
- social state recognition using non-parametric Bayesian learning;
- dialogue modelling using learning automata;
- knowledge representation using ontology learning and reasoning; and
- gesture recognition using AdaBoost, random forests, ordered means-models and hidden Markov models.

The structure of the workshop will consist of individual presentations by authors followed by short question and discussion sections concerning their work. In addition, the workshop features two renowned invited speakers who will present their perspectives on modern frameworks for interactive systems and interactive robots. The workshop will close with a general discussion section that aims to collect and summarise ideas raised during the day (e.g. advances and challenges) and come to a common conclusion.

We are a looking forward to a day of interesting and exciting discussion.

Heriberto Cuayáhuitl
Lutz Frommberger
Nina Dethlefs
Hichem Sahli

(MLIS-2012 organizers)

INVITED TALK

DATA-DRIVEN METHODS FOR ADAPTIVE MULTIMODAL INTERACTION

OLIVER LEMON, HERIOT-WATT UNIVERSITY, UK

How can we build more flexible, adaptive, and robust systems for interaction between humans and machines? I'll survey several projects which combine language processing with robot control and/or vision (for example, WITAS and JAMES), and draw some lessons and challenges from them. In particular I'll focus on recent advances in machine learning methods for optimising multimodal input understanding, dialogue management, and multimodal output generation. I will argue that new statistical models (for example combining unsupervised learning with hierarchical POMDP planning) offer a unifying framework for integrating work on language processing, vision, and robot control.

Prof. Dr. Oliver Lemon leads the Interaction Lab at the school of Mathematical and Computer Sciences (MACS) at Heriot-Watt University, where he is Professor of Computer Science. He works on machine learning methods for intelligent and adaptive multimodal interfaces, on topics such as Speech Recognition, Spoken Language Understanding, Dialogue Management, and Natural Language Generation. He applies this research in Human-Robot Interaction, Technology Enhanced Learning, and situated Multimodal Dialogue Systems.

See: http://www.macs.hw.ac.uk/InteractionLab

INVITED TALK

AUTONOMOUS LEARNING IN INTERACTIVE ROBOTS

JEREMY WYATT, UNIVERSITY OF BIRMINGHAM, UK

In this talk I will give an overview work on learning in robots that have multiple sources of input, and in particular that can use a combination of vision and dialogue to learn about their environment. I will describe the kinds of architectural problems and choices that need to be made to build robots that can choose learning goals, plan how to achieve those goals, and integrate evidence from different sources. To that end I will focus on Dora and George, two robot systems that use natural language to guide their behaviour, developed as part of the CogX project. I will also describe how methods for planning under state uncertainty can be used to drive information gathering and thus learning in interactive robots.

Dr. Jeremy Wyatt leads the Intelligent Robotics Laboratory at Birmingham University, where he is Reader in Robotics and Artificial Intelligence. He is interested in a number of problems, all of which are motivated by the same scientific goal: studying general architectures and methods for learning and reasoning in autonomous agents, especially those with bodies. He has worked on the exploration-exploitation problem in reinforcement learning, the problem of managing diversity in committees of learning machines, cognitive architectures for intelligent robotics, learning of predictions in robot manipulation, planning and learning of information gathering strategies in robots, and on the use of physics knowledge in prediction and estimation in vision.

See: http://www.cs.bham.ac.uk/research/groupings/robotics/

TECHNICAL PAPERS

Machine Learning of Social States and Skills for Multi-Party Human-Robot Interaction

Mary Ellen Foster and **Simon Keizer** and **Zhuoran Wang** and **Oliver Lemon**[1]

Abstract. We describe several forms of machine learning that are being applied to social interaction in Human-Robot Interaction (HRI), using a robot bartender as our scenario. We first present a data-driven approach to social state recognition based on *supervised learning*. We then describe an approach to social interaction management based on *reinforcement learning*, using a data-driven simulation of multiple users to train HRI policies. Finally, we discuss an alternative *unsupervised learning* framework that combines social state recognition and social skills execution, based on hierarchical Dirichlet processes and an infinite POMDP interaction manager.

1 MOTIVATION

A robot interacting with humans in the real world must be able to deal with socially appropriate interaction. It is not enough to simply achieve task-based goals: the robot must also be able to satisfy the social obligations that arise during human-robot interaction. Building a robot to meet these goals presents a particular challenge for input processing and interaction management: the robot must be able to recognise, understand, and respond appropriately to social signals from multiple humans on multimodal channels including body posture, gesture, gaze, facial expressions, and speech.

In the JAMES project[2], we are addressing these challenges by developing a robot bartender (Figure 1) which supports interactions with multiple customers in a dynamic setting. The robot hardware consists of a pair of manipulator arms with grippers, mounted to resemble human arms, along with an animatronic talking head capable of producing facial expressions, rigid head motion, and lip-synchronised synthesised speech. The input sensors include a vision system which tracks the location, facial expressions, gaze behaviour, and body language of all people in the scene in real time, along with a linguistic processing system combining a speech recogniser with a natural-language parser to create symbolic representations of the speech produced by all users. More details of the architecture and components are provided in [3].

The bartending scenario incorporates a mixture of task-based aspects (e.g., ordering and paying for drinks) and social aspects (e.g., managing simultaneous interactions, dealing with arriving and departing customers). For the initial version of the system, we support interactions like the following, in which two customers approach the bar, attract the robot's attention, and order a drink:

Figure 1. The JAMES robot bartender

A customer approaches the bar and looks at the bartender	
ROBOT:	[Looks at Customer 1] How can I help you?
CUSTOMER 1:	A pint of cider, please.
Another customer approaches the bar and looks at the bartender	
ROBOT:	[Looks at Customer 2] One moment, please.
ROBOT:	[Serves Customer 1]
ROBOT:	[Looks at Customer 2]
	Thanks for waiting. How can I help you?
CUSTOMER 2:	I'd like a pint of beer.
ROBOT:	[Serves Customer 2]

In subsequent versions, we will support extended scenarios involving a larger number of customers arriving and leaving, individually and in groups, and with more complex drink-ordering transactions. We are also developing a version of this system on the NAO platform.

2 SOCIAL STATE RECOGNITION

In general, every input channel in a multimodal system produces its own continuous stream of (often noisy) sensor data; all of this data must be combined in a manner which allows a decision-making system to select appropriate system behaviour. The initial robot bartender makes use of a rule-based social state recogniser [10], which infers the users' social states using guidelines derived from the study of human-human interactions in the bartender domain [7]. The rule-based recogniser has performed well in a user evaluation of the initial, simple scenario [3]. However, as the robot bartender is enhanced to support increasingly complex scenarios, the range of multimodal input sensors will increase, as will the number of social states to recognise, making the rule-based solution less practical. Statistical

[1] School of Mathematical and Computer Sciences, Heriot-Watt University, email: {M.E.Foster, S.Keizer, Z.Wang, O.Lemon}@hw.ac.uk

[2] `http://james-project.eu/`

In Proceedings of the ECAI Workshop on Machine Learning for Interactive Systems: Bridging the Gap Between Language, Motor Control and Vision, Montpellier, France, pages 9-11, 2012.

approaches to state recognition have also been shown to be more robust to noisy input [14]. In addition, the rule-based version only considers the top hypothesis from the sensors and does not consider their confidence scores: incorporating other hypotheses and confidence may also improve the performance of the classifier in more complex scenarios, but again this type of decision-making is difficult to incorporate into a rule-based framework.

A popular approach to addressing this problem is to train a supervised classifier that maps from sensor data to user social states. The system that is most similar to our robot bartender is the virtual receptionist of Bohus and Horvitz [1], which continuously estimates the engagement state of multiple users based on speech, touch-screen data, and a range of visual information including face tracking, gaze estimation, and group inference. After training, their system was able to detect user engagement intentions 3–4 seconds in advance, with a low false positive rate. Other recent similar systems include a system to predict user frustration with an intelligent tutoring system based on visual and physiological sensors [8], and a classifier that used body posture and motion to estimate children's engagement with a robot game companion [12].

Applying similar techniques to the robot bartender requires a gold-standard multimodal corpus labelled with the desired state features. (An alternative to using labelled data is explored in work using unsupervised learning methods [13], described in Section 4.) We are currently developing such a corpus based on logs and video recordings from users interacting with the initial robot bartender [3], along with data recorded from human-human interactions in real bars [7]. The state labels capture both general features of multi-party social interaction such as engagement and group membership, as well as domain-specific states such as the phases of ordering a drink. We are also carrying out signal processing and feature extraction on the raw data to turn the continuous, multimodal information into a form that is suitable for supervised learning toolkits such as WEKA [5]. The resulting classifier will be integrated into the next version of the robot bartender, where its output will be used as the basis for decision making by a high-level planner [10] as well as by the POMDP-based interaction manager described below.

3 SOCIAL SKILLS EXECUTION

The task of social skills execution involves deciding what response actions should be generated by the robot, given the recognised current social state as described in the previous section. Such actions include both communicative actions (i.e., dialogue acts, such as greeting or asking a customer for their order), social actions (such as managing queueing), and non-communicative actions (typically, serving a drink); the system must also decide how communicative actions are realised, i.e., which combinations of modalities should be used (speech and/or gestures). This decision-making process should lead to robot behaviour that is both task-effective and socially appropriate. An additional challenge is to make this decision-making robust to the generally incomplete and noisy observations that social state recognition is based on.

Automatic learning of such social skills is particularly appealing when operating in the face of uncertainty. Building on previous work on statistical learning approaches to dialogue management [14], we therefore model social skills execution as a Partially Observable Markov Decision Process (POMDP) and use reinforcement learning for optimising action selection policies. Action selection in our multi-modal, multi-user scenario is subdivided into a hierarchy of three different stages with three associated policies. The first stage is concerned with high-level multi-user engagement management; the second stage involves deciding on response actions within an interaction with a specific user; and the final stage involves multimodal fission, i.e., deciding what combination of modalities to use for realising any such response actions. Each of the policies provides a mapping from states to actions, where the state space is defined by features extracted from the recognised social state.

As in the POMDP approaches to dialogue management, we use simulation techniques for effective policy optimisation. For this purpose, a multi-modal, multi-user simulated environment has been developed in which the social skills executor can explore the state-action space and learn optimal policies. The simulated users in the environment are initialised with random goals (i.e., a type of drink to order), enter the scene at varying times, and try to order their drink from the bartender. At the end of a session, each simulated user provides a reward in case they have been served the correct drink, incorporating a penalty for each time-step it takes them to get the bartender's attention, to place their order and to be served. This reward function is based on the behaviour of customers interacting with the current prototype of the robot bartender [3], who responded most strongly to task success and dialogue efficiency. Policy optimisation in this setting then involves finding state-action mappings that maximise the expected long-term cumulative reward.

Preliminary experiments on policy optimisation have demonstrated the feasibility of this approach in an MDP setup, i.e., under the assumption that the recognised social states are correct. The action selection stages of multi-user engagement and single-user interaction are modelled by a hierarchy of two MDPs, which are optimised simultaneously using a Monte Carlo control reinforcement learning algorithm. The trained strategies perform at least as well as a hand-coded strategy, which achieves a 100% success rate in noise-free conditions when using simulated users which are very patient (i.e., they keep trying to make an order until the session is ended externally by the simulated environment). The trained system starts to outperform the hand-coded system when the simulated users are set to be less patient (i.e., they give up after a maximum number of time-steps) and/or when noise is added to the input.

An important current goal is to make more use of collected human-human and human-machine data to make the user simulation as realistic as possible, and therefore to ensure that the trained social skills executor is more likely to perform well in interaction with real user. A further goal is to explicitly represent the uncertainty underlying the social state recognition process, and to exploit this uncertainty in a POMDP framework for more robust social skills execution.

4 AN UNSUPERVISED FRAMEWORK

As an alternative to the preceding supervised approaches to social state recognition and social skills execution, which require labelled data, we have also developed a non-parametric Bayesian framework for automatically inferring social states in an unsupervised manner [13], which can be viewed as a natural fusion of multimodal observations. This approach makes use of the infinite POMDP method [2], which does not require advance knowledge of the size of the state space, but rather lets the model grow to accommodate the data.

To adapt the infinite POMDP to multimodal interactions, we define a distribution for every observation channel, and let the joint observation distribution be their tensor products, where distributions of different forms can be utilised to capture different representations of observations. For example, the Bernoulli distribution that has a conjugate Beta prior is a natural choice to model binary discrete events,

such as gesture occurrences. When generalised to the multivariate case, it also models the occurrences of events in n-best lists such as ASR hypotheses, where respective Beta distributions can be used conjunctively to draw the associated (normalised) confidence scores. (Although Beta likelihood does not have a conjugate prior, one can either employ Metropolis-Hastings algorithms to seek a target posterior [6], or perform a Bernoulli trial to choose one of its two parameters to be 1 and apply a conjugate Gamma prior for the other one [9].) Finally, to model streams of events, multinomial or multivariate Gaussians can be used to draw the respective discrete or continuous observation in each frame, for which conjugate priors are the well-known Dirichlet distribution and Normal-Inverse-Wishart distribution, respectively.

In addition, to allow the optimised POMDP policy to find a timing solution, and to avoid rapid state switches, we adapt the idea of the "sticky" infinite HMM [4] here as follows. Firstly, state inference is performed for every frame of observations, where "null" actions are explicitly defined for the frames between system actions. Then, transition probabilities depending on the "null" actions are biased on self-transitions using the same strategy as [4], with the assumption that users tend to remain in the same state if the system does not do anything (although the probabilities of implicit state transitions are still preserved). After this, at each timestamp a trained policy either decides on a particular action or does nothing.

Initial experiments have been performed using a human-human interaction corpus from the bartender domain [7]. We employ the forward search method proposed in [2] for iPOMDPs to perform action selection, where a set of models is sampled to compute a weighted-average Q-value, and only a finite set of observations generated by Monte-Carlo sampling are maintained at each node of the search tree. The decisions computed based on the "sticky" infinite POMDP agree with the human actions observed in the corpus in 74% of cases, which outperforms the standard iPOMDP and is comparable to a supervised POMDP trained based on labelled data. Moreover, our system selected many of the correct actions more quickly than the human bartender did [13].

At this stage, our non-parametric Bayesian approach only handles single-user interactions. Multi-party interactions can be addressed by hierarchical action selection [11] with higher-level actions specifying which user to interact with and lower-level actions executing the actual plans, where hierarchical action policies can be trained via reinforcement learning based on simulated user environments. These aspects are our ongoing work, and will be integrated into the next version of the robot bartender system.

5 SUMMARY

We have presented a range of machine learning techniques that we are using to explore the challenges of multi-modal, multi-user, social human-robot interaction. The models are trained on data collected from natural human-human interactions as well as recordings of users interacting with the system. We have given initial results using real data to train and evaluate these models, and have outlined how the models will be extended in the future.

ACKNOWLEDGEMENTS

The research leading to these results has received funding from the European Union's Seventh Framework Programme (FP7/2007–2013) under grant agreement no. 270435, JAMES: Joint Action for Multimodal Embodied Social Systems. We thank our colleagues on the JAMES project for productive discussion and collaboration.

REFERENCES

[1] Dan Bohus and Eric Horvitz, 'Dialog in the open world: platform and applications', in *Proceedings of the 2009 International Conference on Multimodal Interfaces (ICMI-MLMI 2009)*, pp. 31–38, (November 2009).

[2] Finale Doshi-Velez, 'The infinite partially observable Markov decision process', in *Proceedings of NIPS*, (2009).

[3] Mary Ellen Foster, Andre Gaschler, Manuel Giuliani, Amy Isard, Maria Pateraki, and Ronald P. A. Petrick, "'Two people walk into a bar": Dynamic multi-party social interaction with a robot agent'. In submission, 2012.

[4] Emily B. Fox, Erik B. Sudderth, Michael I. Jordan, and Alan S. Willsky, 'An HDP-HMM for systems with state persistence', in *Proceedings of ICML 2008*, (2008).

[5] Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H. Witten, 'The WEKA data mining software: an update', *SIGKDD Explorations Newsletter*, **11**(1), 10–18, (November 2009).

[6] Michael Hamada, C. Shane Reese, Alyson G. Wilson, and Harry F. Martz, *Bayesian Reliability*, Springer, 2008.

[7] Kerstin Huth, *Wie man ein Bier bestellt*, Master's thesis, Universität Bielefeld, 2011.

[8] Ashish Kapoor, Winslow Burleson, and Rosalind W. Picard, 'Automatic prediction of frustration', *International Journal of Human-Computer Studies*, **65**(8), 724–736, (2007).

[9] Tomonari Masada, Daiji Fukagawa, Atsuhiro Takasu, Yuichiro Shibata, and Kiyoshi Oguri, 'Modeling topical trends over continuous time with priors', in *Proceedings of the 7th International Symposium on Neural Networks*, (2010).

[10] Ronald P. A. Petrick and Mary Ellen Foster, 'What would you like to drink? Recognising and planning with social states in a robot bartender domain', in *Proceedings of the 8th International Conference on Cognitive Robotics (CogRob 2012)*, (July 2012).

[11] Joelle Pineau, Nicholas Roy, and Sebastian Thrun, 'A hierarchical approach to POMDP planning and execution', in *ICML Workshop on Hierarchy and Memory in Reinforcement Learning*, (2001).

[12] Jyotirmay Sanghvi, Ginevra Castellano, Iolanda Leite, André Pereira, Peter W. McOwan, and Ana Paiva, 'Automatic analysis of affective postures and body motion to detect engagement with a game companion', in *Proceedings of the 6th International Conference on Human-Robot Interaction (HRI 2011)*, pp. 305–312, (March 2011).

[13] Zhuoran Wang and Oliver Lemon, 'Time-dependent infinite POMDPs for planning real-world multimodal interactions', in *ESSLLI Workshop on Formal and Computational Approaches to Multimodal Communication*, Opole, Poland, (2012).

[14] S. Young, M. Gašić, S. Keizer, F. Mairesse, B. Thomson, and K. Yu, 'The Hidden Information State model: a practical framework for POMDP based spoken dialogue management', *Computer Speech and Language*, **24**(2), 150–174, (2010).

Fast Learning-based Gesture Recognition for Child-robot Interactions

Weiyi Wang and **Valentin Enescu** and **Hichem Sahli** [1]

Abstract. In this paper we propose a reliable gesture recognition system that could be run on low-level machines in real-time, which is practical in human-robot interaction scenarios. The system is based on a Random Forest classifier fed with Motion History Images(MHI) as classification features. To detect fast continuous gestures as well as to improve the robustness, we introduce a feedback mechanism for parameter tuning. We applied the system as a component in the child-robot imitation game of ALIZ-E project.

1 INTRODUCTION

Human gesture and movement recognition plays an important role in robot related interaction scenarios. The system we describe in this paper can detect dynamic gestures in video sequences using a machine learning approach. So far, four types of gestures defined in the Simon game (children - robot imitation game) of ALIZ-E project are recognized: Left-Arm-Up, Right-Arm-Up, Left-Arm-Down and Right-Arm-Down [9], while extension to other more complicated ones is trivial, provided training data is available. Moreover, the system provides the probabilities of each pre-defined gesture, which is useful to tell "how good you are" in the children-robot game scenario.

The main contribution of this work lies in proposing a reliable gesture recognition system based on motion history features and a random forest classifier, with low computational requirements. This makes it fit for low-end machines, such as various robots where the processors are not as powerful as normal computers. Temporal segmentation is not necessary as we continuously calculate the MHIs w.r.t. each received frame, then feed them to the classifier, while a feedback mechanism is introduced between the two modules.

Owing to its importance in human-computer interactions, plenty of research work has been done on this topic so far. Mitra et al. [7] conducted a literature survey, in which some widely used mathematical models as well as tools or approaches that helped the improvement of gesture recognition were discussed in details. [8] widened the survey of human action recognition and addressed some challenges such as variations in motion performance and inter-personal differences. Usually, the recognition procedure is computationally intensive and time consuming. To address this issue, [6] developed a real-time hand gesture recognizer running on multi-core processors and [2] implemented a GPU-based system which also runs in real-time. Unlike our approach, both these methods require a specific hardware setup.

Motion history images (MHI) represent a view-based temporal template method which is simple but robust in representing movements [1]. To employ MHI as classification features without feature selection or dimensionality reduction (which is time consuming), we need a classifier that could handle highly-dimensional features effectively. According to [5], random forests [4] have the best overall performance in this situation. The combination of these two approaches makes our system efficient and provides reliable recognition results.

[1] Dept. of Electronics and Informatics (ETRO), Vrije Universiteit Brussel (VUB), Brussels, Belgium. Email: {wwang, venescu, hsahli}@etro.vub.ac.be

2 IMPLEMENTATION

The system structure is depicted in Figure 1.

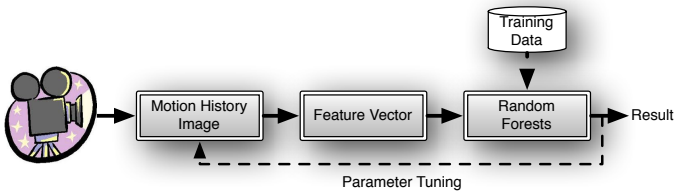


Figure 1. System structure

2.1 Feature extraction

Our processing loop starts by cropping the video images to the upper body area using the face detection approach provided in the OpenCV library. Since the face detection is a time consuming process and in our game scenario the child's body does not move too much, we only perform the face detection once and keep constant the cropping parameters for the ensuing video images. In other more general cases, it is easy to run it in a separate thread or periodically update the cropping area after a certain number of frames.

A motion history image is computed as soon as the system receives a new captured image from the camera or a video file. Refer to [3] for the details of motion history images calculation. We resize the MHI to a proper resolution such that the resulting feature vector is large enough to contain sufficient information and small enough to be easily handled by the classifier. After experimentation, we set the MHI width and height to be 80 and 60, respectively. Hence the dimension of feature vector is 4800.

One important parameter of MHI is the duration time, which determines the span of time before the motion "fades out" in MHI after a gesture is performed. We set it at two seconds based on the assumption that this is the time span of one gesture.

In Figure 2, on the left side, one can see the original captured image (320 * 240) from the camera/video file and, on the right side, the

In Proceedings of the ECAI Workshop on Machine Learning for Interactive Systems: Bridging the Gap Between Language, Motor Control and Vision, Montpellier, France, pages 13-15, 2012.

resized motion history image (80 * 60) for that frame. In the captured image, the red rectangle indicates the cropping area based on the position of the detected face region.

2.2 Classification

Training data was prepared in the form of labeled video clips performed by several subjects. We use the last frame of the motion history images of each video file to derive the features as described in Section 2.1. After the feature vectors are obtained from all video clips, they are fed to a Random Forest algorithm as a batch to train the classifier.

During the recognition phase, we compute the feature vector of each frame, which further serves as input for the classifier. The Random Forest algorithm is a voting-based classifier whereby the decision is made by collecting the votes from the trees in the forest. Before the final decision, we evaluate the votes for each class. Only when one class received the most votes and they exceed a certain threshold (i.e., a percentage of the number of trees in the forest), it is considered as a recognized result. In this way, the static gesture and irrelevant movements will be distinguished as "Idle" or, say, "Unknown" type. This threshold needs to be set properly: low values will make the system too sensitive to any kind of movement, thereby increasing the number of false alarms; high values will be too strict for discrimination. An optimal value for the threshold can be found by plotting the histogram of votes for the four defined gesture classes as well as the histogram for the "Idle/Unknown" class, and then taking the value at the boundary between the two histograms (or the middle value of the overlapping part) as threshold. Temporal segmentation is unnecessary as the MHI features contain temporal information and are continuously (i.e., at each frame) computed and fed to the classifier, which will make a decision whenever a certain class receives enough votes.

Due to the inherent character of random forests, it is feasible to derive the probabilities of classes in real-time – simply divide the number of votes of one class by the number of trees in the forest. Those values are important in our child-robot imitation game scenario to indicate "how good you are".

Figure 2. Motion history features of continuous gestures without feedback tuning of duration time

2.3 Feedback

In the right image of Figure 2, one can see that a "Right-Arm-Up" gesture immediately followed by a "Left-Arm-Up" gesture results in the MHI featuring both of them in the same time. This phenomenon hurdles the classification in taking the right decision as the two different gestures can not be properly discriminated, thereby preventing the reliable recognition of quick gesture sequences.

To solve this problem, we devised a feedback mechanism for the duration time of MHI calculation (see Section 2.1). As soon as the system detects a certain gesture, the system decreases the duration time to a minimum value such that the trace of last gesture fades out immediately. After a certain period of time (e.g., 500 ms), this parameter is increased back to the normal value to enable MHI to capture the trace of gestures lasting as long as two seconds.

3 RESULTS

To assess experimentally the system performance, we recorded 80 video clips by two subjects as training data, i.e., 10 repetitions for each gesture. Then other six subjects were asked to perform each gesture again 10 times to test the system. At all times, the subject was asked to return to the neutral gesture with two hands positioned around the waist. Moreover, we asked the same subjects to perform some randomly irrelevant movements apart from the four gestures (e.g., both arms up/down, waving etc.), which are considered as belonging to the "Unknown" class.

We set the number of trees in random forests as 200 and the size of the randomly selected subset of features at each tree node is set to be 100. The tests were run on single core at 1.6 GHz and the system reached an average speed of 29.6 frames per second. The confusion matrix is presented in Table 1. Left-Up and Right-Up were sometimes recognized as their "Down" counterparts, as the motion regions of Up/Down gestures are partially overlapped.

We compared our method with the one proposed in [2] that used AdaBoost classifier fed with optical flow features to recognize similar gestures (punch-left, punch-right, sway, wave-left, wave-right, waves, idle), in which they achieved 20.7 frames per second with GPU acceleration and a recognition accuracy of 87.3%, at the same resolution (320 * 240).

Table 1. Confusion matrix for the four gestures classes and the unknown class. Rows represent the true movements, and columns represent the numbers as well as percentage output by the system.

	L-U	L-D	R-U	R-D	Unknown
Left-Up	57	1	0	0	2
	95%	1.7%	0%	0%	3.3%
Left-Down	0	58	0	0	2
	0%	96.7%	0%	0%	3.3%
Right-Up	0	0	55	2	3
	0%	0%	91.7%	3.3%	5%
Right-Down	0	0	0	59	1
	0%	0%	0%	98.3%	1.7%
Unknown	3	3	2	1	51
	5%	5%	3.3%	1.7%	85%

4 CONCLUSION

We have proposed a gesture recognition algorithm that can achieve a good accuracy in real-time, even on the low-end machines. We have applied it in a gesture imitation game between children and the humanoid NAO robot. It has high potential to be run on the on-board processor of the NAO robot, due to the low computation requirements. As the motion history image features do not contain direction information about the movements, we plan to enhance the feature vectors with the MHI gradient calculation to improve recognition rates.

ACKNOWLEDGEMENTS

The research work reported in this paper was supported by the EU FP7 project ALIZ-E grant 248116, and by the CSC-VUB scholarship grant. We also would like to thank the referees for their comments and suggestions, which helped improve this paper considerably.

REFERENCES

[1] Md. Atiqur Rahman Ahad, J. K. Tan, H. Kim, and S. Ishikawa, 'Motion history image: its variants and applications', *Mach. Vision Appl.*, **23**(2), 255–281, (March 2012).
[2] Mark Bayazit, Alex Couture-Beil, and Greg Mori, 'Real-time motion-based gesture recognition using the gpu', in *IAPR Conference on Machine Vision Applications (MVA)*, (2009).
[3] A. F. Bobick and J. W. Davis, 'The recognition of human movement using temporal templates', *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **23**, 257–267, (2001).
[4] L. Breiman, 'Random forests', *Machine Learning*, **45**(1), 5–32, (2001).
[5] R. Caruana, N. Karampatziakis, and A. Yessenalina, 'An empirical evaluation of supervised learning in high dimensions', in *Proceedings of the 25th international conference on Machine learning*, ICML '08, pp. 96–103, New York, NY, USA, (2008). ACM.
[6] T. Ike, N. Kishikawa, and B. Stenger, 'A real-time hand gesture interface implemented on a multi-core processor', in *MVA*, pp. 9–12, (2007).
[7] S. Mitra and T. Acharya, 'Gesture recognition: A survey', *IEEE Transactions on Systems, Man, and Cybernetics, Part C*, **37**(3), 311–324, (2007).
[8] R. Poppe, 'A survey on vision-based human action recognition', *Image and Vision Computing*, **28**(6), 976–990, (June 2010).
[9] R. Ros, M. Nalin, R. Wood, P. Baxter, R. Looije, Y. Demiris, T. Belpaeme, A. Giusti, and C. Pozzi, 'Child-robot interaction in the wild: advice to the aspiring experimenter', in *Proceedings of the 13th international conference on multimodal interfaces*, ICMI '11, pp. 335–342, New York, NY, USA, (2011). ACM.

Using Ontology-based Experiences for Supporting Robot Tasks - Position Paper

Lothar Hotz and **Bernd Neumann** and **Stephanie von Riegen** and **Nina Worch**[1]

Abstract. In this paper, we consider knowledge needed for interaction tasks of an artificial cognitive system, embodied by a service robot. First, we describe ideas about the use of experiences of a robot for improving its interactivity. Our approach is based on an multi-level ontological representation of knowledge. Thus, ontology-based reasoning techniques can be used for exploiting experiences. A robot interacting as a waiter in a restaurant scenario guides our considerations.

1 Introduction

For effective interactions of an artificial cognitive system in a non-industrial environment, not every piece of knowledge can be manually acquired and modeled in advance. Learning from experiences is one way to tackle these issues. Experiences can be defined as "an episodic description of occurrences and own active behavior in a coherent space-time segment". Experiences can be used for future situations by generalization. Generalizations (or *conceptualizations*) build the basis for further interactions and possible implications. Such interactions then constitute the current source for experiences which again can be integrated and combined with existing conceptualizations.

For approaching this task of experience-based learning, we consider a service robot acting in a restaurant environment, see the simulated environment in Figure 1.

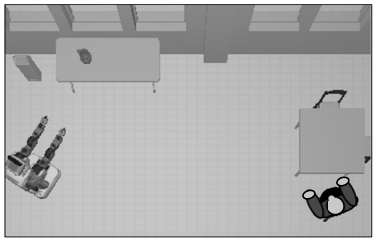


Figure 1: Simulation example: A robot serves a cup to a guest.

In such an environment, domain-specific objects, concepts, and rooms have to be represented. Objects can e.g. be used for a certain purpose and can have impacts on the environment. Different types of relationships between objects have to be considered: taxonomical on the one hand and spatial or temporal relationships on the other hand. Terminological knowledge about dishes, drinks, meals as well as actions and possible occurrences is needed. Areas which may contain served orders (at a table) may be distinguished from seating areas. To perform complex tasks, we consider the interaction that is needed to serve a guest. Moreover, to learn a model for such a process, we examine experiences that result from performing such operations, and investigate how to generalize them.

Our approach is based on ontological knowledge, which comprises models, presented in Section 2 and experiences, introduced in Section 3. Section 4 presents possible generalizations that lead to new conceptualizations in form of new ontological models. A short overview of the architecture of our approach will be given in Section 5 and a discussion of our approach finalizes the paper in Section 6.

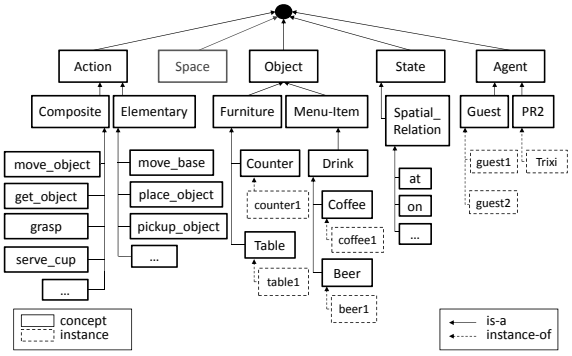


Figure 2: Taxonomical relations of actions and physical objects

2 Ontology-Based Approach

Due to the service domain as well as the inherent interaction with the environment and thereby with agents within, a continuous need of knowledge adjustment to such a dynamic application area is essential. In our approach, an ontology represents the knowledge an agent needs for interacting. This knowledge covers concepts about objects, actions, and occurrences in a TBox (like cup, $plate$, $grasp$, $serve\_cup$ etc.) as well as concrete instances of such concepts in an ABox [1]. Taxonomical relations (depicted in Figure 2) and compositional relations, presented in 3 are essential means for modeling.

A complex activity like serve_cup is decomposed into finer activities until we get a sequence of elementary actions, that the robot can execute directly. Not only these taxonomical and compositional relations, but temporal constraints represent the possible order of actions, like e.g. for the action serve_cup: "Take coffee mug from counter and place it on tray. Go to table, look for guest and place coffee mug in front of guest." Technically, we model binary relations with OWL2[2]

[1] HITeC e.V. c/o Fachbereich Informatik, Universität Hamburg, Germany email: {hotz, neumann, svriegen, worch}@informatik.uni-hamburg.de

[2] `www.w3.org/TR/owl2-overview`

In Proceedings of the ECAI Workshop on Machine Learning for Interactive Systems: Bridging the Gap Between Language, Motor Control and Vision, Montpellier, France, pages 17-19, 2012.

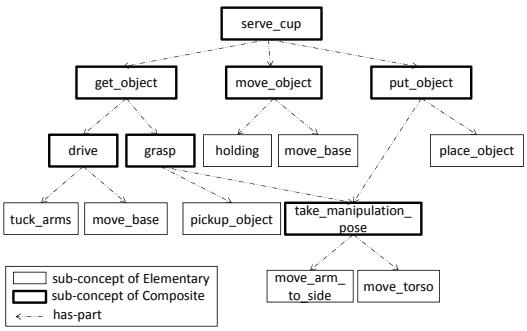


Figure 3: Compositional relations of actions

and n-ary relations, like temporal constraints for complex actions, with SWRL[3], see [2].

3 Experiences

Experiences must be gained by the robot, while the robot is accomplishing a task and will be processed afterwards. In our ontological approach, experiences are also represented as ABox instances (see Figure 4). Thereby, experiences can be represented at all abstraction levels: the complete compositional structure of robot activities, including motions, observations, problem solving and planning, and inter-agent communication. Furthermore, relevant context information, like description of static restaurant parts and initial states of dynamic parts, as well as an indicator of the TBox version are used during experience gaining.

Parallel to robot's interactions, raw data is gathered in subsequent time slices (*frames*) for a certain time point. From these slices, time segments (ranges) of object occurrences and activities are computed (e.g. *grasp* in Figure 5). Such an experience is passed on to a generalization module which integrates the new experience with existing ones.

The initial experience is based on an action of the handcrafted ontology. The outcome of the generalization module will be integrated in the ontology. In general, experiences are gained continuously, thus during every operation, but are dedicated to a goal. We reckon with a manageable number of experiences, because of the successive execution of goals.

Since the experiences are relevant to specific goals, we do not distinguish between experiences that are more important than others at present. But according to "background noise" in the scenery (like a dog walking past during a serve action) some parts of experience might be more significant than others. The accomplishment of this circumstance is presented in the following Section 4.

4 Generalization

We consider an incremental generalization approach, where an initial ontology is extended based on experiences using suitably chosen generalization steps. New experiences are integrated into existing conceptualizations in a cyclic manner. Table 1 shows typical generalization steps based on Description Logic (DL) syntax. Those can be standard DL services (like subsumption of concepts or realization of instances) and non-standard services (like least common subsumers (LCS) [1]). As an example, consider two experiences gained serving coffee to guests, depicted in Figure 4. In principle, all instance tokens are candidates for generalization, e.g. *table1* to *table*. Depending on the commonalities and differences between distinct experiences, however, promising generalizations can be selected, e.g. *coffee1, coffee2* $\rightarrow$ *coffee* $\rightarrow$ *drink*. In order to deal with new situations the robot extends its competence.

Over-generalization, e.g. generalizing *coffee* not to *drink* but to *thing* can be avoided by applying the LCS, by the use of the LCS *drink* is selected. However, when the integration of new concepts is impossible over-generalization can not be prevented.

Generalization Path: from $\rightarrow$ to	Reasoning Service
instance $\rightarrow$ set of instances	realization
instance $\rightarrow$ closest named concept	realization
instance $\rightarrow$ concept expression	realization
set of instances $\rightarrow$ concept expression	realization
concept $\rightarrow$ superconcept	subsumption
set of concepts $\rightarrow$ concept expression	LCS
role cardinality range $\rightarrow$ larger role cardinality range	range union
role filler concept restriction $\rightarrow$ generalized role filler concept restriction	LCS
numerical ranges $\rightarrow$ larger numerical ranges	range union

Table 1: Ontology-based generalizations and their computation through reasoning services

In Section 3 we raised the issue of experience parts that might be more significant than others, on the example of a dog walking past during a serve activity. We cover this circumstance by integrating cardinalities to mark that a dog may appear but it is not mandatory.

In addition to ontological generalization, temporal and spatial constraints can be generalized. Figure 5 presents an example for a temporal generalization. Quantitative temporal orderings by concrete time points are generalized to qualitative temporal relations.

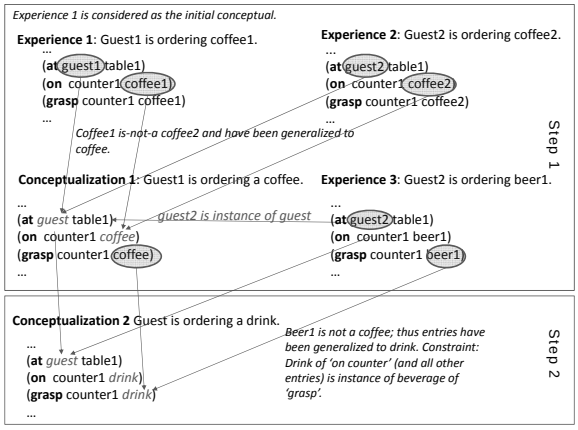


Figure 4: Example for creating conceptualizations from two experiences, or one experience and a conceptualization

5 Architecture

Experiences do not contain only observed data, like perceived actions, objects and relational information but also occurrences and robot's states. These experience contents are gathered by the components presented in Figure 6. Information on object detections (like the identification of *counter1*) and spatial relations (e.g. *(on counter1 coffee1)*) are released by the *object publisher*. The *action publisher*

[3] www.w3.org/Submission/SWRL/

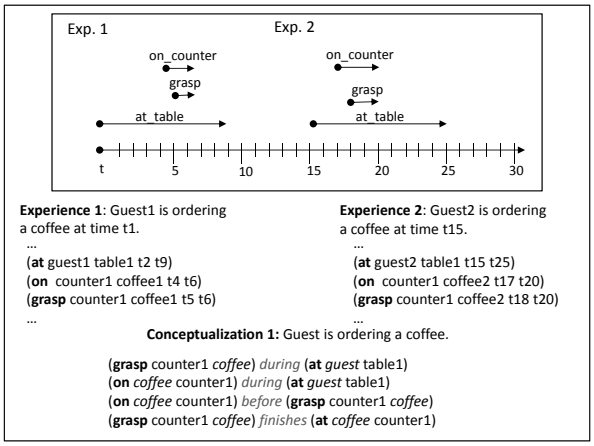


Figure 5: Temporal generalizations preserving temporal order

exports performed action informations, like *(grasp counter1 coffee1)*. Extremity informations of the robot, like the position of the torso or of an arm are published by the *actuator monitor*. These outputs are gathered by the *integration manager*. This manager provides the *experience manager* with this content. The *reasoner* offers reasoning services and the *learning* component generalizes current experiences (in the homonymous module) or complex scene examples to new models. All kinds of knowledge about objects, actions, occurrences and the environment are described in the *ontology*, which will be extended based on experiences made by the robot during it's processing. The *experience database* is a storage location, hold available already gained experiences in a specific format.

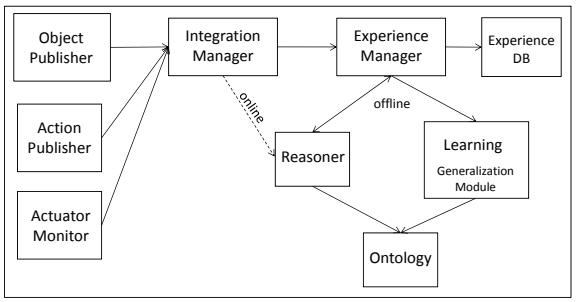


Figure 6: Architecture overview

6 Discussion

In this paper, we presented an ontology-based method for dealing with robot interaction tasks in a dynamic application area. The ontology model provides a central framework for all task relevant knowledge. By successively extending a hand-coded ontology through generalizing from experiences, a learning scheme is realized. [3] presents a similar approach for rudimentary actions like grasping or door opening, we consider aggregated actions like serving a cup to a guest. However, in both cases, experiences provide the basis for refinement of actions.

Representing a robot's knowledge in a coherent way by an ontology, we are able to use existing ontology-based reasoning techniques like DL services. Ontology alignment can also be applied to integrate experiences obtained with different TBoxes (e.g. differing because of new conceptualizations). Similar methods must be applied for generalizing temporal and spatial experiences. Although we propose continuous gathering of experiences, one might as well consider scenarios building the source for an experience that have explicit start and end points (similar to [3]).

Some parts of an experience may be more significant than others, it may be useful to focus on experiences which were made in respect to a specific goal. Furthermore, not every detail should be the subject of generalization, the temporal order or equality of instances in a complex action have to be preserved (more concrete: the cup that is served should be the same cup that was taken from the counter before).

With the aggregation of occurrences, states and elementary actions (covering also agent interactions) to composites and the expansion of knowledge via experience gaining an extension of the interaction ability with the environment and people within is achieved.

ACKNOWLEDGEMENTS

This work is supported by the RACE project, grant agreement no. 287752, funded by the EC Seventh Framework Program theme FP7-ICT-2011-7.

REFERENCES

[1] F. Baader, D. Calvanese, D. McGuinness, D. Nardi, and P. Patel-Schneider, *The Description Logic Handbook*, Cambridge University Press, 2003.

[2] W. Bohlken, B. Neumann, L. Hotz, and P. Koopmann, 'Ontology-Based Realtime Activity Monitoring Using Beam Search', in *ICVS 2011*, ed., J.L. et al. Crowley, LNCS 6962, pp. 112–121, Heidelberg, (2011). Springer Verlag.

[3] Alexandra Kirsch, *Integration of Programming and Learning in a Control Language for Autonomous Robots Performing Everyday Activities*, Ph.D. dissertation, Technische Universität München, 2008.

A Corpus Based Dialogue Model for Grounding in Situated Dialogue

Niels Schütte and **John Kelleher** and **Brian Mac Namee**[1]

Abstract. Achieving a shared understanding of the environment is an important aspect of situated dialogue. To develop a model of achieving common ground about perceived objects for a human-robot dialogue system, we analyse human-human interaction data from the Map Task experiment using machine learning and present the resulting model.

1 Introduction

The problem of achieving a shared understanding of the environment is an important part of situated dialogue. It is of particular importance in a situated human-robot dialogue scenario. The application scenario for our work is that of a semi-autonomous tele-operated robot that navigates some environment, and is controlled through dialogue by a remote human operator. The robot uses a camera to perceive the environment and sends the video feed on to the operator.

The operator gives instructions to the robot using natural language. For example, the operator may instruct the robot to perform a move by giving the instruction "Go through that door", using an object from the environment as a landmark (LM). The success of such instructions depends on whether or not the operator and the robot agree about their understanding of the objects in the environment. The robot will not be able to execute the move-instruction felicitously if it has not recognized the object the operator is referring to, and is therefore not aware of its presence. Another possible problem could arise if the robot has recognized the presence of the object, but has not classified it in the same category as the operator, and for example thinks is a large box or a window. This may be the case due to problems with the robot's objects recognition mechanisms.

It is therefore necessary that the participants reach a mutual understanding about what they perceive. We suggest that this problem can be understood as a part of the grounding problem, i.e. the achieving of a **common ground** [3] in a dialogue.

The problem of grounding in general and in human-computer dialogue in particular has been addressed by a number of authors (e.g. [6]), but we are not aware of work that addresses the problem we described. With this work we do not intend to provide a comprehensive discussion of grounding but to focus on a quantitative analysis of a specific and small area of grounding in a visual context. We additionally hope to use the techniques explored in this work as the basis of further work in our domain. In some sense the problem we address is also related to the symbol grounding problem [4] because it deals with achieving an agreement about how to treat sensoric perception in the linguistic domain. However, in this work we focus on grounding in the sense we initially discussed.

[1] Dublin Institute of Technology, Ireland, email: niels.schutte@student.dit.ie, john.d.kelleher@dit.ie,brian.macnamee@dit.ie

Our overarching interest is in recognising the occurrence of such problematic situations and in identifying strategies to avoid and resolve them, taking into account the characteristics of the robot domain such as multimodal interaction or object recognition mechanisms that can be primed. In this work we focus on the aspect of grounding of newly introduced objects from the environment. We also plan to use an approach that is based in quantitative corpus analysis rather than single examples.

We are not aware of any corpus data that directly relates to the phenomenon in question. We instead use data from the map HCRC Map Task Corpus [1] which we believe contains similar phenomena.

The paper is structured as follows. In Section 2 we introduce the data set we use in this work and the steps we took to extract data. In Section 3 we describe the steps we took to analyse the data and our preliminary results. In Section 4 we introduce the model we developed based on our observations. In Section 5 we discuss our results and in Section 6 we describe our planned next steps.

2 Data

The map task corpus contains data from interactions involving two participants who worked together to solve a task involving a visual context. The task consisted of the following: The participants were issued separate maps. On one participant's map (we call this participant the **instruction giver** or **g** in the following) a target route was marked, the other participant's map (the **instruction follower** or **f**) did not contain the route. Figure 1 contains an example of such a map pair. The instruction giver was asked to describe the route on their map to the instruction follower, who was asked to reproduce the described route on their own map. The participants were allowed to engage in dialogue, but were not able to see the other participant's map.

In total the corpus contains 128 dialogues that use 16 different instruction giver/follower map pairs. Each dialogue was annotated for a number of phenomena. For our experiment we were interested in the *dialogue move* annotations because they provided us with a good level of abstraction over the structure and contents of the dialogues, and the landmark reference annotations because they indicated to us when participants were talking about objects in the visual context. The dialogue move set used in the dialogue move annotations is detailed in the corresponding coding manual [2]. All data related to the dialogue transcripts and their annotation could be efficiently accessed through a query-based tool and an API [5].

What makes this data set interesting for us is that there are a number of differences between the maps used by the instruction giver and follower. For example, landmarks that are present on one map may be missing on the other map, or landmarks on one map may be re-

In Proceedings of the ECAI Workshop on Machine Learning for Interactive Systems: Bridging the Gap Between Language, Motor Control and Vision, Montpellier, France, pages 21-26, 2012.

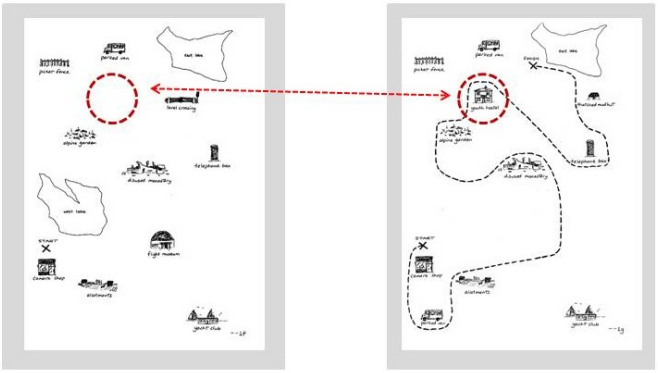


Figure 1. An instruction follower/giver map pair. Highlighted is an example of a landmark that is present on one map and missing on the other one.

placed on the other map by different but similar landmarks (e.g. a landmark called "white water" on the instruction giver's map may be called "rapids" on the follower's map). We assume that the way the participants handled these problems would be analogous to the way problems arising from different perception of objects in the human-robot dialogue scenario could be handled.

We were interested in instances where an object was for the first time referred to in the dialogues. Our approach was to detect instances in the dialogue where a landmark is introduced for the first time, and to record how the introduction is performed and what the reaction to the introduction consists in. We did this in the following way.

We took each dialogue separately and extracted all references to landmarks. We then sorted these references by the landmark they referred to and ordered them by their time stamp. We then selected the earliest reference. This gave us the first mention of a landmark. Using this information we could then extract the utterance that contained the reference, as well as preceding and succeeding utterances as context. In total we extracted 1426 initial references. We expected that landmarks would be treated differently based on whether they were (a) initially visible to both participants (b) visible to only either instruction giver or follower, or (c) visible to both but with a difference. This meant each landmark would fall into one of four conditions:

Condition 1: The landmark appears on both maps in the same place.
Condition 2: A landmark appears on both maps in the same place, but there is a difference between the landmarks.
Condition 3: The landmark is on the instruction giver's map, but not on the follower's map.
Condition 4: The landmark is on the instruction follower's map, but not on the instruction giver's map (basically the inverse of Condition 3).

We determined for each landmark on the maps into which condition it fell by manually comparing the instruction giver/follower map pairs. In Table 1 we show how many instances we found for each condition. As we can see the majority of landmarks are shared between the participants. Most of the remaining landmarks fall either into Condition 3 or 4, while only a small number falls into Condition 2.

The participants typically approached the task in such a fashion that the instruction giver visited the landmarks step by step as indicated by the route on their map and instructed the follower to draw the route based on the landmarks. This meant that initiative in the dialogue was primarily one-sided and with the instruction giver.

Condition	Count	Proportion
1	787	55.0%
2	69	4.8%
3	302	21.1%
4	268	18.8%

Table 1. Number of landmark instances per condition.

As mentioned previously our goal for this work is to model the grounding of newly introduced objects from the visual context. We base our approach on extracting sequences of dialogue moves that occur in the context of where a new object is introduced in the dialogue, and then extracting general strategies from them. We focused on one specific type of sequence, namely sequences that were started with a *query_yn*-move that contained an initial reference to a landmark and finished with an *instruct*-move

A *query_yn*-move is defined as a move that asks the other participant a question that implies a "yes" or "no" answer. We assumed that if a *query_yn*-move contained a reference to a landmark, it would most likely be about the landmark and could be seen as an attempt by the speaker to find out whether or not the other participant had the landmark on their map. We manually checked some example moves and this appeared to be a reasonable assumption. An *instruct*-move is a move in which the speaker asks the other participant to perform some action. Usually this move refers to instructions to the other participant to draw a stretch of the route. To be able to better distinguish between different instruction giving strategies, we split the annotated *instruct*-moves into two more specific moves: the *instruct_LM*-move refers to *instruct*-moves that contain a references to a landmark and the *instruct_NOLM*-move refers to *instruct*-moves that do not contain a reference to a landmark. We based this distinction on the landmark reference annotations that were contained in the corpus data. In general we assumed that the *instruct_LM*-moves used the contained landmark as a point of reference, while the *instruct_NOLM*-move did not use a landmark as a point of reference, but contained only directional move instructions (e.g. "go to the left and slightly upwards").

We assumed that these sequences would typically comprise a piece of dialogue that consisted of the following elements:

- The introduction of a landmark by the instruction giver.
- The reaction of the follower, possibly a counter reaction and the grounding of the landmark.
- The instruction move using either the grounded landmark or some alternative strategy that has been decided upon due to the outcome of the grounding process.

Figure 2 contains an example of a typical piece of dialogue we captured.

We decided to focus on Condition 1 and Condition 3 landmarks at this stage of the work. We expected that Condition 4 landmarks would exhibit fundamentally different phenomena in the dialogue because the landmarks in this condition were only visible to the follower, and could therefore only be introduced by the instruction follower. We also excluded Condition 2 landmarks at this stage because of the small number of available examples. We found 290 sequences for the Condition 1 domain and 129 sequences for the Condition 3 domain. We may revisit the other conditions at a later stage under a different perspective.

g:	erm have you got a collapsed shelter *(query_yn)*
f:	yes i do *(reply_y)*
g:	right *(acknowledge)*
g:	you've to go up north and then round the collapsed shelter *(instruct_LM)*

Figure 2. An example of a query-instruction dialogue.

To be able to distinguish between successful strategies and unsuccessful strategies, we decided to annotate for each landmark along a route on the instruction giver's map how well the route on the instruction follower's map had been reproduced when the route visited the corresponding landmark. We asked the annotators to compare each map produced in a dialogue by a instruction follower with the map given to the instruction giver, and give a judgement for each landmark along the route. We allowed three possible categories:

Good: The route on the follower's map matches the route on the instruction giver's map.

Ok: The route on the follower's map roughly matches the route on the instruction giver's map, but it is apparent that the follower did not take special care to take the landmark into account when they drew the route.

Bad: The route on the follower's map does not match the route on the instruction giver's map at all.

We then assigned to each sequence the value annotated for the landmark mentioned in the sequence. This way we got an indication of how successful each sequence was.

We used this information to filter the set of instances and only used those that had been annotated as "good". This left us with 271 Condition 1 instances (93.1% of the original Condition 1 instances) and 90 Condition 3 instances (69.7% respectively).

In the following section we describe the steps we took in analysing the data.

3 Analysis

Our aim is to create a model that explains the process of grounding in the map task domain and that we can adapt to drive the same process in our human-robot domain.

Our first goal of the analysis was to determine whether there were any dominating structures in the dialogue move sequences that we could later on use to develop dialogue strategies. As a second goal we wanted to see if there were other, less dominant, structures that occurred with some consistency and might be appropriate to specific situations. Our third goal was to analyse the structures and to see if we could develop plausible assumptions about why these structure come about, i.e. a model of the underlying information state of the dialogue.

Due to the large number of examples, it was not feasible to perform a manual analysis. We therefore used machine learning to extract structure.

To gain a general overview of the sequences and their commonalities we decided to create a graph representation of the move sequences in the domain that conflated sequences where they were similar and branched out where they diverged.

To this purpose we added to each move its position in the sequence as an index. This means, the sequence

g_query_yn → *f_reply_y* → *g_acknowledge* → *g_instruct_LM*

will be represented as

g_query_yn_0 → *f_reply_y_1* → *g_acknowledge_2* → *g_instruct_LM*

This was based on the idea that if two sequences contained the same move at the same position, they would be similar at this point. We did not add an index the final *instruct*-move of each sequence because due to the manner in which sequences were created each sequences ends with one.

We then created a graph where each node represents one of the indexed dialogue moves. Two nodes n_1 and n_2 were connected if any sequence contained an instance where the move corresponding to n_1 was directly followed by the move corresponding to n_2. Each arc was labelled with the number of times such an instance occurred in the sequences. Figure 3 shows the graph created for Condition 1 and the graph for Condition 3 (note that for readability and ease of presentation we omitted arcs with counts less than 5 in the first graph and counts of 3 in the second graph).

A general observation we can draw from the graphs is that landmark based move instructions in both conditions are more frequently used than non-landmark based ones. However, it appears that the tendency to use non-landmark based instructions is stronger in the Condition 3 domain. This is plausible, because in the Condition 3 domain the instruction giver cannot use the landmark they initially asked about as a point of reference, and may therefore be more likely to switch to a direction based strategy.

We then used the sequence analysis tool of the SAS Enterprise Miner 6.2[2] to detect typical sequences from the set of observation sequences. Sequences 1-4 in Table 2 are some selected interesting sequences from the Condition 1 domain that we produced in this step. Sequences 1-5 in Table 3 are the interesting sequences we extracted from the Condition 3 domain.

It appears that there are clear differences in the detected sequences, namely that the Condition 1 domain strongly features Yes-No queries that receive a positive answer, while the Condition 3 domain features queries with negative answers. We also get complete sequences that start with a *query_yn*-move and end with an *instruct*-move. They describe the most typical complete sequences. When we compare the detected sequences with the graphs, we discover that the longer sequences in fact correspond to paths through the graph that have high count figures along the arcs. But we also see that there are alternative paths which have lower count figures but could nevertheless be important enough to be worth modelling.

As a general observation we can see that *reply*-moves are often responded to with either an *acknowledge*-move, a *ready*-move or a combination of both. However, it appears that these moves may also be omitted. To gain a better understanding of the domain, we repeated the sequence detection process, but this time we removed *acknowledge*- and *ready*-moves from the sequences because we suspected that they introduced noise that might prevent other significant sequences from being detected. This resulted in Sequence 5 in Table 2 for the Condition 1 domain and Sequence 6 and 7 in Table 3 for the Condition 3 domain. These sequences do not contain complete se-

[2] *SAS Enterprise Miner*, Version 6.2, www.sas.com/technologies/analytics/datamining/miner/

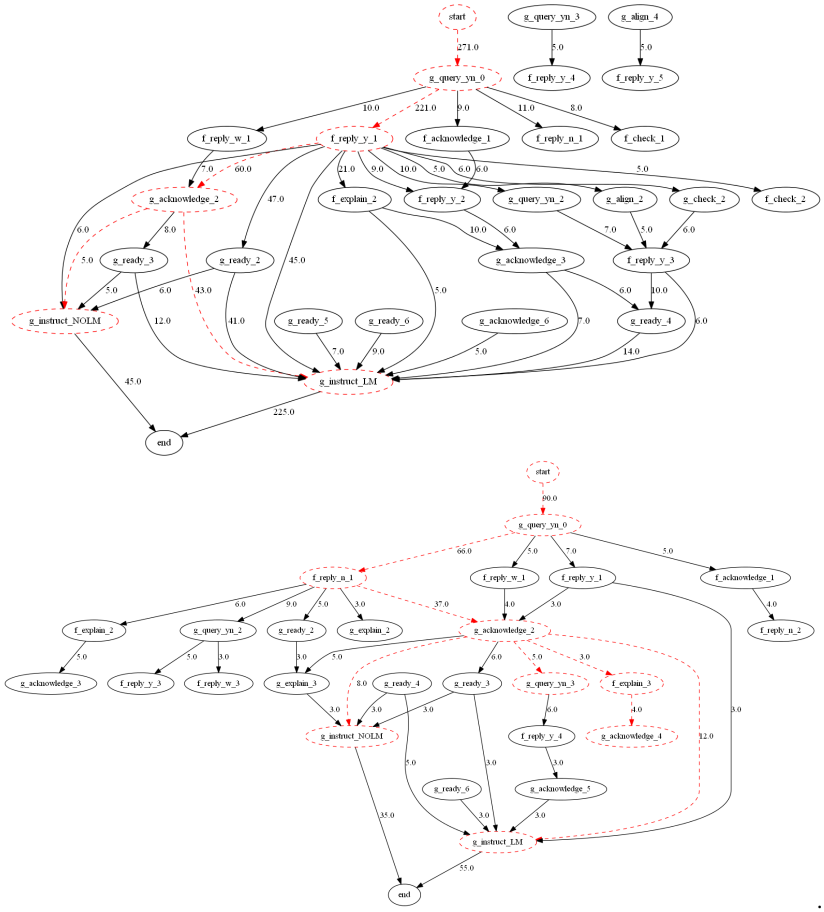


Figure 3. Dialogue move graphs for Condition 1 and Condition 3. The nodes and arcs that are highlighted with dashed red lines represent sequences that are covered by the model introduced in Section 4.

Number	Length	Sequence
1	2	g_query_yn → f_reply_y
2	3	g_query_yn → f_reply_y → g_instruct_LM
3	4	g_query_yn → f_reply_y → g_acknowledge → g_instruct_LM
4	3	g_query_yn → f_reply_y → g_instruct_NOLM
5	4	g_query_yn ⇒ f_reply_y ⇒ f_explain ⇒ g_instruct_LM

Table 2. Interesting sequences from the Condition 1 domain.

Number	Length	Sequence
1	2	g_query_yn → f_reply_n
2	3	g_query_yn → f_reply_n → g_instruct_LM
3	3	g_query_yn → f_reply_n → g_instruct_NOLM
4	4	g_query_yn → f_reply_n → g_acknowledge → g_instruct_LM
5	4	g_query_yn → f_reply_n → g_acknowledge → g_instruct_NOLM
6	3	g_query_yn → f_reply_n → g_query_yn
7	3	g_query_yn → f_reply_n → f_explain

Table 3. Interesting sequences from the Condition 3 domain.

quences but important sub-sequences which would be important for our model.

3.1 Condition 1

For Condition 1, the dominant structure appears to be one where the follower responds positively to the request and the instruction giver then issues a landmark based instruction, or alternatively a non-landmark based instruction. This is supported by the Sequences 2-5 in Table 2.

Another important structure appears to be one where the follower responds positively and then adds an *explain*-move (this is exemplified by the Sequence 6 from Table 2). The instruction giver then proceeds with the move instruction as normal. We manually inspected some sample sequences and concluded that this *explain*-move may serve one of three purposes:

- It may confirm the landmark by repeating it.
- It may mention an additional landmark that is close to the intended landmark.
- It may describe the location of the intended landmark in relation to the current position.

We believe that this extra move generally serves as an additional grounding step. The second move is of course context dependent because it requires that there is a suitable landmark available.

3.2 Condition 3

In Condition 3 we can identify a dominant structure where the follower responds negatively to the query, and the instruction giver then issues a move instruction (Sequences 2-5 in Table 3). Another structure appears to be one where the instruction giver, instead of issuing the instruction immediately, issues another query (Sequence 6). We interpret this as the instruction giver testing out a different landmark for their instruction.

As another possibility, the follower may also offer an *explain*-move after the negative reply. We examined the moves manually and determined that they either serve to mention explicitly that the follower does not see the landmark in question or to offer an alternative landmark (Sequence 7). We show an example for such a sequence in Figure 4. The instruction giver may or may not use a landmark in the *instruct*-move that concludes the sequence. We examined some samples from the dialogues and found that in the cases where a landmark is used, it is a landmark that has been discussed in the dialogue immediately prior to the current exchange, and is therefore still salient.

g	i sp– i don't suppose you've got a graveyard have you
f	ehm no
g	no right
f	got a fast running creek and canoes and things

Figure 4. An example of a dialogue where the follower offers an alternative landmark.

Based on these observations we developed a model of how these grounding sequences can be performed in a human-robot dialogue system is presented in the following section.

4 Model

In the previous section we introduced an analysis of the structures encountered in the dialogues, and made some suggestions about the underlying reasons for these structures. Based on this, we are now going to present a finite state model that can be used by a dialogue system to model grounding. This model is shown in Figure 5. In our model, a robot **system** is engaged in a dialogue with a human **operator**. Both operator and system have access to a shared visual context.

In this model we take an object as grounded from the perspective of the system if the system perceives the object, knows that the other participants perceives the object in the same way, and knows that the other participants knows that the system perceives the object.

The model uses an information state consisting of the following components:

G: the set of grounded objects.
D: the set of "discarded" objects (should therefore be avoided for shared reference e.g. because an attempt to ground them has failed).
i: an object that has been referred to by the other participant (an abstract object reference that may match objects in the visual context).
f: an object in the visual context of the system that is in the focus of attention.
df: an object that the system has declared is in focus.
dn: an object that the system has declared it does not perceive.

The model is intended as a sub-part of a larger model that controls the system's dialogue. We assume that this larger model maintains sets equivalent to G and D that can be used to instantiate the model. The model triggers when the operator produces a *query_yn* about a landmark. The system takes the reference and attempts to resolve it in its visual context. If it succeeds, we branch into the left side of the model (we base this part of the model on our analysis of the Condition 1 domain). The object that had been found is put into focus, and the system produces a *reply_y* to indicate it has found the object. The object is not grounded yet, but the system has declared that it perceives the object. If the operator then produces an *acknowledge*-move, the object is added to the set of grounded objects.

If the system is unable to resolve the reference in the first step, we go into right side of the graph (this part of the model is based on our analysis of the Condition 3 domain). The system produces an *reply_n*-move to indicate this fact. We represent this in the state by storing the object in *dn*. If the operator reacts with an *acknowledge*-move, we add the object to the set of discarded objects. If the operator poses a new query at this point, we model this as a return to the first state with a new intended object while retaining the set of shared and discarded objects.

As we discussed in the previous section, it occurs in some cases that the follower suggests an alternative landmark. We suggest to model this in the following way: The system may check if there is an object available in the place where it expects the object introduced by the operator to be, e.g. based on direction expressions in the introduction. It then expresses this with an *explain*-move. If the operator acknowledges this by making an *acknowledge*-move, it is entered into the set of grounded objects G.

Based on our observations, basically at any point after the first response the operator can be expected to produce an instruction, either using a landmark or not using a landmark. We believe that the grounding state of the object used in the instruction determines how

appropriate the use of the object as a landmark is. In particular we believe that an object that is in focus and in the common ground is most acceptable (this would be an object that has undergone the process on the left side of the diagram). Slightly less acceptable would be an object that has been focused, but is not yet in the common ground (an object that has undergone the process on the left side except for the final *acknowledge*-move). It is also possible to use an object that is in G, the set of grounded objects, but not focused (this corresponds to the case of the instruction giver using an object that has been introduced prior to the current sub-dialogue), but we believe that this would be a less preferred option.

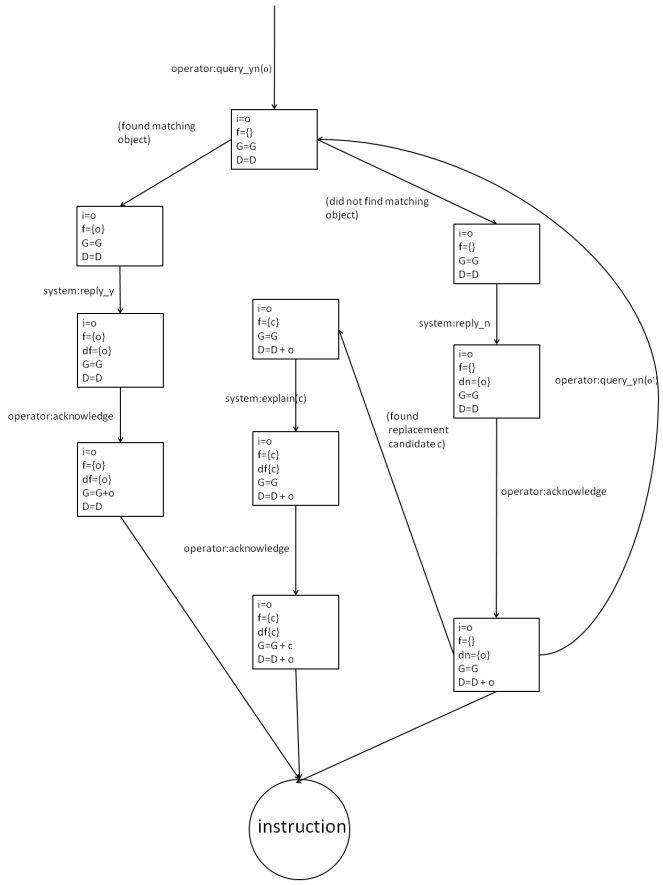


Figure 5. The finite state model. The boxes represent states, while the arcs represents actions by the system or the operator. Each state is annotated with current configuration of the information state of the system.

5 Discussion

We performed a quantitative analysis of corpus data and extracted typical interaction sequences. The findings are certainly not surprising or counter to what other works about grounding describe. We nevertheless believe that they are relevant because they are based on data rather than a mostly manual analysis. This analysis also shows us that parts of the domain can be captured, while highlighting those parts of the domain that are not covered and remains to be investigated. As mentioned earlier, Figure 3 provides an overview of possible interactions in the domain. We highlight the nodes and arcs that are covered by our model. We calculated that for the Condition 1 domain, about 18% of the observed cases are covered in and about 28% for the Condition 3 domain. These number are low, but they still represent the major observed structures in the domain. In addition to that it appears that optional *ready* and *acknowledge* moves introduce variation that is hard to capture with a model as simple as ours (for example, collapsing some of the *ready* and *acknowledge* acts increases the coverage in the Condition 1 domain to about 53%).

6 Future Work

There are a number of possible future directions for this work. Our main line of interest will be to set up an evaluation system that we can use to examine how well the strategies we developed work in an application scenario. We based parts of this work on manually annotated information which will not be available in an online application scenario. We will therefore in the near future focus on using machine learning based tools to replicate equivalent information. We believe that the data we have available at this stage will be useful as training data for those components. In addition to spoken dialogue, we are also considering to investigate other modalities such as markup information in the video displayed to the operator.

There are also some possible topics left to address within this data set, such as the conditions we have not addressed in this work, and other types of interactions. In particular we are also interested in problems such as error recovery and clarification after a problematic reference.

REFERENCES

[1] A. Anderson, Bard Bader, M., Boyle E., G. M. E., Doherty, S. Garrod, S. Isard, J. Kowtko, J. McAllister, J. Miller, C. Sotillo, H. S. Thompson, and R. Weinert, 'The HCRC map task corpus', *Language and Speech*, **34**(4), 351–366, (1992).

[2] Jean Carletta and Amy Isard, 'HCRC dialogue structure coding manual', Technical report, Centre, University of Edinburgh, (1996).

[3] Herbert H. Clark and Edward F. Schaefer, 'Contributing to discourse', *Cognitive Science*, 259–294, (1989).

[4] Stevan Harnad. The symbol grounding problem, 1990.

[5] Jonathan Kilgour and Jean Carletta, 'The nite xml toolkit: demonstration from five corpora', in *Proceedings of the 5th Workshop on NLP and XML: Multi-Dimensional Markup in Natural Language Processing*, NLPXML '06, pp. 65–68, Stroudsburg, PA, USA, (2006). Association for Computational Linguistics.

[6] David R. Traum, 'Computational models of grounding in collaborative systems', in *Working Papers of the American Association for Artififial Intelligence Fall Symposium on Psychological Models of Communication*, eds., Alain Brennan, Susan E.and Giboin and David R. Traum, pp. 124–131, Menlo Park, California, (1999). American Association for Artificial Intelligence.

Hierarchical Multiagent Reinforcement Learning for Coordinating Verbal and Non-verbal Actions in Robots

Heriberto Cuayáhuitl[1] and **Nina Dethlefs**[2]

Abstract. This paper proposes an approach for learning to coordinate verbal and non-verbal behaviours in interactive robots. It is based on a hierarchy of multiagent reinforcement learners executing verbal and non-verbal actions in parallel. Our approach is evaluated in a conversational humanoid robot that learns to play Quiz games. First experimental results show evidence that the proposed multiagent approach can outperform hand-coded coordinated behaviours.

1 Introduction

Multiagent Reinforcement Learning is used to build autonomous agents that learn their behaviour from a shared environment [3]. In the case of cooperative Reinforcement Learning (RL) agents, they use the same reward function in order to optimize a joint goal [2, 12, 13, 10]. Recent research on interactive systems using machine learning has experienced important progress in the optimization of their conversational behaviours (e.g. confirmation, clarification and/or negotiation dialogues), where the RL framework has been an attractive alternative to hand-coded behaviours for the design of optimized dialogue agents. However, although important progress has been made for speech-based interactive systems, less progress has been made on optimizing both verbal and non-verbal behaviours in a unified way. Instead, both types of behaviours are often modelled independently [1, 15, 14, 8], without the aim to jointly achieve a goal as is the case in human interaction, where verbal and non-verbal behaviours are tightly coupled [16].

In this paper, we propose an approach based on hierarchical multiagent RL for optimizing the coordination of verbal and non-verbal behaviours. In this approach, one agent optimizes verbal behaviour, while another (simultaneously) optimizes non-verbal behaviour so as to align with the non-verbal actions of a human user. As a result, the joint action-selection of the RL agents represents the optimized coordination of both behaviours. We present preliminary results suggesting that this form of joint optimization is a promising and principled alternative to non-joint approaches and can equip robots with a more natural way of coordinating and adapting their multimodal actions.

2 Proposed Learning Approach

To achieve scalable dialogue optimization, we cast interaction control as a discrete-time Multiagent Semi-Markov Decision Process (MSMDP) $M = <S, \vec{A}, T, R, L, F>$ that is characterized by the following elements: (a) a finite set of states S; (b) a finite set of joint actions $\vec{A} = (A^v, A^{nv})$ executed in parallel, where A^v are verbal actions and A^{nv} are non-verbal actions; (c) a stochastic state transition function $T(s', \tau|s, \vec{a})$ that specifies the next state s' given the current state s and joint action $\vec{a} = (a^v, a^{nv})$, where τ denotes the number of time-steps taken to execute joint action $\vec{a}$ in state s; (d) a reward function $R(s', \tau|s, \vec{a})$ that specifies the reward given to the agent for choosing joint action $\vec{a}$ when the environment makes a transition from state s to state s'; (e) a language L that is represented as a context-free grammar (CFG) to represent relational tree-based representations as described in [4]; and (f) a stochastic model transition function $F = P(m'|m, s)$ that specifies the next model or subtask m' given model m and state s. This last element allows the user to navigate more flexibly across the available sub-dialogues [5].

We distinguish two types of actions: (i) single-step joint actions[3] corresponding to verbal actions such as 'greeting' or 'ask question' and non-verbal actions such as 'head nodding' or 'lift right arm', and (ii) multi-step joint actions corresponding to sub-dialogues or conjunctions of single-step joint verbal and non-verbal actions. In addition, we treat each multi-step joint action as a separate MSMDP.

We decompose an MSMDP into multiple MSMDPs that are hierarchically organised into X levels and Y models per level. The indices (i, j) only identify a unique subtask (i.e. MSMDP) in the hierarchy, they do not specify the execution sequence of subtasks which is learnt by the RL agent, where $j \in \{0, ..., X-1\}$ and $i \in \{0, ..., Y-1\}$. Thus, a given MSMDP in the hierarchy is denoted as $M^{(i,j)} = <S^{(i,j)}, \vec{A}^{(i,j)}, T^{(i,j)}, R^{(i,j)}, L^{(i,j)}, F^{(i,j)}>$. Notice that each MSMDP is a multi-decision maker for verbal and non-verbal actions, hence the term 'multiagent'. The solution to a Multiagent Semi-Markov Decision Process is an optimal policy $\pi^{*(i,j)}$, which is a mapping from environment states $s \in S$ to single- or multi-step joint actions $\vec{a} \in \vec{A}$. The goal of an MSMDP is to find a function denoted as $\pi^{*(i,j)}(s)$ that maximizes the cumulative reward of each visited state. The optimal policy for each learning agent in the hierarchy is defined by $\pi^{*(i,j)}(s) = \arg\max_{\vec{a} \in \vec{A}^{(i,j)}} Q^{*(i,j)}(s, \vec{a})$, where the optimal action-value function $Q^{*(i,j)}(s, \vec{a})$ specifies this cumulative reward for executing joint action $\vec{a}$ in state s and then following policy $\pi^{*(i,j)}$. We apply the HSMQ-Learning algorithm [9, 6] to cooperatively induce such a hierarchy of multiagent policies based on long-term cumulative rewards across policies.

3 Experimental Setting

To test our approach for generating coordinated joint actions and compare it with non-coordinated baselines, we use a robot dialogue

[1] German Research Center for Artificial Intelligence (DFKI), Saarbrücken, Germany, email: hecu01@dfki.de
[2] Heriot-Watt University, Edinburgh, Scotland, email: n.s.dethlefs@hw.ac.uk

[3] We assume that the execution of single-step joint actions terminates at the same time, which involves a non-verbal action to wait for the verbal one to terminate, or vice versa. Other ways of termination, where agents behave more autonomously but still in a coordinated way, are left as future work.

In Proceedings of the ECAI Workshop on Machine Learning for Interactive Systems: Bridging the Gap Between Language, Motor Control and Vision, Montpellier, France, pages 27-29, 2012.

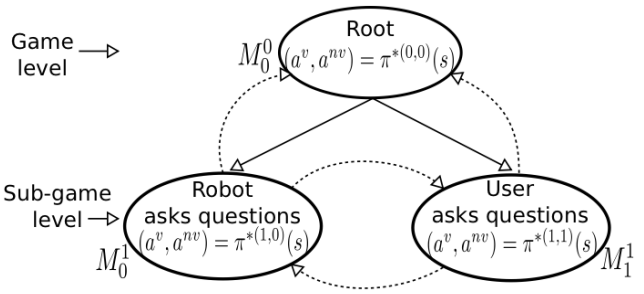


Figure 1. Hierarchy of joint agents for our robot in the Quiz domain. Whilst straight lines denote strict hierarchical control, dashed lines denote less strict control for more flexible interaction across sub-dialogues [5].

system playing Quiz games. In this domain, the robot can ask the user questions, or vice-versa, the user can ask the robot questions. Both user and robot communicate with verbal and non-verbal actions and our aim is to coordinate the robot's non-verbal actions with its verbal actions and simultaneously align them with the user's non-verbal actions to show individualised adaptation. Our system has been implemented using the Nao humanoid robot (see dialogue in Table 2).

We use the hierarchy of dialogue agents shown in Figure 1. Table 1 shows the set of state variables for our system, each one modelled as a discrete probability distribution with predefined parameters. Dialogue and game features are included to inform the agent of situations in the interaction. The set of verbal actions (80 in total) consists of meaningful combinations of speech act types and associated parameters.[4] The set of non-verbal actions (20 in total) consists of predefined body movements.[5] We constrained the actions per state based on the CFGs $L^{(i,j)}$, so that only a subset of joint actions was allowed per dialogue state (constraints omitted due to space). This reduces the state-action space from 10^{12}, using a propositional representation enumerating all variables and values, to only 10^4.

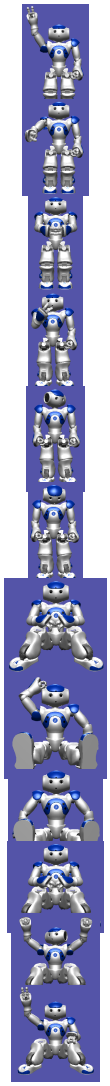
The global **reward function** aims for interactions that encourage to play, get as many correct answers as possible, and imitate the user's non-verbal actions. It is defined by the following rewards for choosing action a in state s: $+10$ for answering a question correctly or reaching a terminal state (in which the user will be prompted to play again), -10 for remaining in the same state (i.e. $s_{t+1} = s_t$ or $s_{t+1} = s_{t-1}$), $+1$ for imitating a non-verbal action, and 0 otherwise. The **multimodal user simulation** used a set of user dialogue acts as responses to the system dialogue acts (Footnotes 4-5). They used predefined probability distributions for modelling verbal and non-verbal interactions: $P(a^{v,usr}|a^{v,sys})$ and $P(a^{nv,usr}|a^{nv,sys})$, with errors based on an equally distributed speech and gesture recognition error rate of 20%. The recognition confidence scores were generated from beta probability distributions with parameters ($\alpha = 2$, $\beta = 6$) for bad recognition and ($\alpha = 6$, $\beta = 2$) for good recognition [4].

[4] **Verbal Single-Step Actions**: Speech Act Types={Salutation, Request, Apology, Confirm, Accept, SwitchRole, Acknowledgement, Provide, Stop, Feedback. Express, Classify, Retrieve, Provide.} × Parameters={Greeting, Closing, Name, PlayGame, Asker, KeepPlaying, GameFun, StopPlaying, Play, NoPlay, Fun, NoFun, GameInstructions, StartGame, Question, Answers, CorrectAnswer, IncorrectAnswer, GamePerformance, Answer, Success, Failure, GlobalGameScore, ContinuePlaying}

[5] **Non-Verbal Single-Step Actions**={Hello, Bye, HandShake, NodYes, NodNo, Success, Failure, OpenRightArm, OpenLeftArm, SitDown, StandUp, SeatedWithExtendedLegs, SeatedWithCrossedLegs, Thinking, ScratchingHead, StandingWithCrossedArms, StandingWithArmsBack, StandingWithArmsHeadBack, Wait, None.}

State Variable	Values
Salutation	none, greeting, withName, regreeting, closing
UserName	unknown, filled, known
ConfScore	null, 0.1, 0.2, 0.3, 0.4, 0.5, ... , 0.9, 1.0
Confirmed	null, no, yes
PlayGame	unknown, no, yes, ready
Instructions	unprovided, provided
Asker	unknown, robot, user
QuizGame	unplayed, playing, semiplayed, played, interrupted, keepPlaying, stopPlaying
GameFun	unknown, no, yes
GameOver	no, yes
GameInstructions	unprovided, provided
QuestionState	null, unknown, unasked, askedWithAnswers, askedWithoutAnswers, reaskedWithAnswers, reaskedWithoutAnswers, confirmed
AnswerState	unanswered, unclassified, correct, incorrect, unknown
MaxQuestions	no, yes
GameScore	unknown, good, bad
GlobalGameScore	null, unprovided, provided
ExpressedScore	no, yes

Table 1. State variables for the Quiz dialogue system, where combinations of variable-value pairs define situations (states) in the interaction used by the reinforcement learning dialogue agents for joint action-selection.



Hi, my name is NAO, what is your name?
I am Charles

Nice to meet you Charles.
Do you want to Play a Quiz game?
Sure

One of us asks questions and offers possible answers.
The other answers by selecting one of those answers.
Do you want to ask me questions or shall I ask you?
You ask me
The first question is. What is a typical magic instrument?
Your choices are: One, guitar. Two, drums.
Three, violin. Four, flute. What is your choice?
Violin

That is not quite right. Do you have another choice?
Maybe flute

Good, that is correct.
Now I want to ask you

Let's do it like that. What is the first question?
What does water become at the temperature of zero degrees?
Your choices are: One, condensation. Two, ice. Three, steam.

I choose ice.
That is correct.

Okay, ask me another question.
I want to stop playing.

Did you like playing the Quiz Game?
I did

I am glad to hear that.

It was nice playing with you, see you soon. Bye!

Table 2. Illustrative multimodal dialogue exhibiting non-verbal actions (left) and verbal actions (right). User responses shown in italics. The robot's images were generated with the Choregraphe tool from `aldebaran.com`

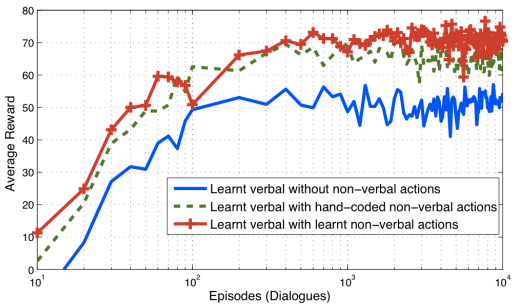


Figure 2. Average reward (10 runs) of joint action learners. Settings: $\alpha = 100/(100+\tau)$, γ of .99, ϵ-Greedy, $\epsilon = .01$, initial Q-values$= 0.01$.

4 Experimental Results

We trained our agents, and compared their performance in terms of *dialogue reward* against two baselines; see Figure 2. One baseline uses *learnt verbal actions without non-verbal actions* (solid blue line), and the other baseline uses *learnt verbal actions with hand-coded non-verbal actions* (dashed green line). The latter baseline included intuitive joint actions such as <*Salutation(Greeting),Hello*> or <*Feedback(CorrectAnswer),NodYes*>. Results from the last 1000 episodes show that our multiagent approach (red crossed line) outperforms its counterparts (blue and green lines) by 27% and 8% in terms of average reward, respectively. We can draw the following preliminary conclusions. While the low performance of the verbal-only baseline most likely results from its lack of non-verbal expressiveness (and therefore lack of positive rewards for imitating the user), the difference between the jointly learnt and hand-coded policies is most likely related to adaptiveness. While the hand-coded policy relies on intuitive combinations of verbal and non-verbal actions, users differ with respect to their individually preferred combinations. Coordinating verbal and non-verbal actions jointly based on imitation of the user's gestures, therefore leads to a higher degree of individualised adaptation and higher rewards.

As a consequence of these results, we will investigate two hypotheses in future research: (1) a humanoid robot that only speaks but does not move has a lower perceived performance than a robot that combines verbal with non-verbal actions; and (2) a humanoid robot that does not learn to coordinate its verbal with non-verbal actions in an adaptive fashion is perceived as having a lower performance than a robot that learns to coordinate both types of actions. An advantage of learning to coordinate verbal with non-verbal actions is that the robot can exhibit different behaviours for different users. Future work may also investigate how coordinated verbal and non-verbal behaviour may affect task success or user satisfaction.

5 Conclusion and Future Work

We have described an approach for optimizing the behaviour of robot dialogue systems by applying and extending a hierarchical RL framework to support multiagent decision making of verbal and non-verbal actions in a coordinated and adaptive way. To evaluate, we have incorporated our methods into a robot dialogue system that learns to play Quiz games. Although preliminary, experimental results make our approach look promising by combining the benefits of (a) predefined state-action spaces, (b) scalable policy learning, (c) joint and coordinated action section, and (d) opportunities for online learning.

We argue that those features, with a special focus on online learning, represent an interesting direction to train robots' behaviour, so that they can learn how to coordinate their actions in an adaptive fashion while interacting with users. The next step towards this is to train our simulations and MSMDPs (online) from real human-robot interactions to validate our results. We would like to optimize turn-taking for more natural and efficient interactions. Another step is a comparison with other hierarchical learning algorithms [11] using function approximation. We also would like to extend our joint learning agents with adaptive verbalizations [7], where each MSMDP in our hierarchy of agents would have three agents, one for dialogue management, one for language generation, and one for non-verbal behaviour.

6 Acknowledgments

This research was funded by the European FP7 programmes under grant agreements ICT-248116 (ALIZ-E) and 287615 (PARLANCE).

REFERENCES

[1] Dan Bohus and Eric Horvitz, 'Facilitating Multiparty Dialog with Gaze, Gesture, and Speech', in *ICMI-MLMI*, p. 5, (2010).
[2] Craig Boutilier, 'Sequential Optimality and Coordination in Multiagent Systems', in *International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 478–485, (1999).
[3] L. Busoniu, R. Babuska, and B. De Schutter, 'A Comprehensive Survey of Multiagent Reinforcement Learning', *IEEE Transactions on Systems, Man, and Cybernetics*.
[4] H. Cuayáhuitl, 'Learning Dialogue Agents with Bayesian Relational State Representations', in *IJCAI Workshop on Knowledge and Reasoning in Practical Dialogue Systems, Barcelona*, pp. 9–15, (Jul 2011).
[5] H. Cuayáhuitl and I. Kruijff-Korbayová, 'An Interactive Humanoid Robot Exhibiting Flexible Sub-Dialogues.', in *North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT), Montreal, Canada*, (Jun 2012).
[6] H. Cuayáhuitl, S. Renals, O. Lemon, and H. Shimodaira, 'Evaluation of a Hierarchical Reinforcement Learning Spoken Dialogue System', *Computer Speech and Language*, **24**(2), 395–429, (2010).
[7] N. Dethlefs and H. Cuayáhuitl, 'Hierarchical Reinforcement Learning for Adaptive Text Generation', in *International Conference on Natural Language Generation (INLG)*, Dublin, Ireland, (Jul 2010).
[8] N. Dethlefs, V. Rieser, H. Hastie, and O. Lemon, 'Towards Optimising Modality Allocation for Multimodal Output Generation in Incremental Dialogue', in *ECAI Workshop on Machine Learning for Interactive Systems: Bridging the Gap Between Language, Motor Control and Vision (ECAI-MLIS), Montpellier, France*, (Aug 2012).
[9] T. Dietterich, 'An Overview of MAXQ Hierarchical Reinforcement Learning', in *Symposium on Abstraction, Reformulation, and Approximation (SARA)*, pp. 26–44, (Jul 2000).
[10] M. Ghavamzadeh and S. Mahadevan, 'Hierarchical Multiagent Reinforcement Learning', *Journal of Autonomous Agents and Multi-Agent Systems*, **13**(2), 197–229, (2006).
[11] Bernhard Hengst, 'Hierarchical Reinforcement Learning', in *Encyclopedia of Machine Learning*, 495–502, (2010).
[12] M. Lauer and M. Riedmiller, 'An Algorithm for Distributed Reinforcement Learning in Cooperative Multi-agent Systems', in *International Conference on Machine Learning (ICML)*, pp. 535–542, (2000).
[13] M. Lauer and M. Riedmiller, 'Reinforcement Learning for Stochastic Cooperative Multi-Agent-Systems', in *Intl. Confrence on Autonomous Agents and Multi-Agent Systems (AAMAS)*, pp. 1516–1517, (2004).
[14] V. Rieser and O. Lemon, 'Learning and Evaluation of Dialogue Strategies for New Applications: Empirical Methods for Optimization from Small Data Sets', *Computational Linguistics*, **37**(1), 153–196, (2011).
[15] A. C. Tenorio-Gonzalez, E. F. Morales, and L. Villaseor-Pineda, 'Dynamic Reward Shaping: Training a Robot by Voice', in *Ibero-American Conference on AI (IBERAMIA), Bahía Blanca, Argentina*, (Nov 2010).
[16] A. Yamazaki, K. Yamazaki, M. Burdelski, Y. Kuno, and M. Fukushima, 'Coordination of Verbal and Non-verbal Actions in Human-Robot Interaction at Museums and Exhibitions', *Journal of Pragmatics*, **42**(9), 2398–2414, (2010).

Towards Optimising Modality Allocation for Multimodal Output Generation in Incremental Dialogue

Nina Dethlefs, Verena Rieser, Helen Hastie and Oliver Lemon[1]

Abstract. Recent work on incremental processing in interactive systems has demonstrated that incremental systems can gain higher responsiveness and naturalness than their non-incremental counterparts and are better perceived by human users. This paper presents a first investigation, based on a proof-of-concept study, into how multimodal information presentation in incremental dialogue systems can contribute towards more efficient and smooth interactions. In particular, we focus on how a combination of verbal and non-verbal output generation can help to reduce the need for self-corrections in a system that has to deal with continuous updates of input hypotheses. We suggest to use Reinforcement Learning to optimise the *multimodal output allocation* of a system, i.e. the idea that for every context, there is a combination of modalities which adequately communicates the communicative goal.

1 Introduction

Traditionally, the smallest unit of processing in interactive systems that triggers a processing module into action has been a complete user utterance. While this facilitates processing and system design, it can lead to inflexible turn-taking and stilted interactions. In contrast, interactive systems with incremental processing align with human-like turn-taking behaviour by defining the *micro-turn* as the smallest unit of processing, which can be seen as the smallest part of an utterance that can be mapped to a dialogue act. This allows them to process input and plan output in parallel and to explore a range of discourse phenomena that occur naturally in human discourse, but that have so far been absent from interactive systems. Among these are backchannel generation, handling of user and system barge-ins, as well as corrections of generated output based on changed user or system knowledge. Several studies have shown that such phenomena can improve the user experience with an interactive system; see e.g. [22, 4] for incremental dialogue management, [18, 8] for turn-taking, [2, 23] for incremental automatic speech recognition, [12, 17, 24, 3] for incremental NLG, and [28] for a study on the impact of real-time feedback on user behaviour. Very recently, incremental processing has also been applied to the information presentation (IP) phase of interactive systems, where it has been combined with machine learning techniques to optimise the timing and order of IP [7] and the timing and occurrence of barge-ins and backchannels [6].

An important advantage resulting from the use of incremental processing is the increased awareness that NLG modules gain of their own generation process: they are able to monitor their own output and, if necessary, e.g. due to updated information coming in from the dialogue manager, modify or self-correct it. Such updates may be necessary in cases where user input hypotheses change during generation (or dialogue processing). As such, incremental NLG has to solve a trade-off between higher system reactiveness versus potentially disturbing self-corrections.

This paper argues that a possible remedy to this problem lies in the combination of different modalities, for example, speech and visual displays on a mobile device. Such multimodality may present a subtle way of communicating the system's current best input hypothesis to the user (and thereby give them a chance to correct it) without mistakenly acting upon it and causing a disruption or delay to the interaction. This hypothesis is based on previous work which has shown that multimodal output generation can increase system robustness to speech recognition errors [10] and decrease user cognitive load [15]. Previous work by [16] has also shown that allowing users to modify their search queries by combing speech and text input can significantly facilitate mobile search in noisy environments.

In this paper, we investigate a model of automatic output generation optimisation that uses *Reinforcement Learning* (RL) to maximise the expected return for the problem of *multimodal allocation* [1], i.e. how to combine output modalities so that they adequately convey a communicative goal in a given context. We present preliminary results from a proof-of-concept study in the domain of restaurant recommendations that compare the *task ease* achieved by our system and a number of hand-crafted baselines in simulated interactions. We discuss the possible advantages and disadvantages of our proposed method with respect to incremental interactive systems in *hands-free, eyes-free* mobile applications.

2 Multimodal Information Presentation

Previous work on multimodal information presentation has investigated rule-based user-tailored content selection [27] and supervised re-ranking techniques [11] for multimodal generation, as well as hierarchical Reinforcement Learning techniques for multimodal dialogue management [20, 5]. However, none of these earlier approaches has considered how multimodal information presentation can be integrated into an incremental model of dialogue processing.

In the following, we extend an earlier model for multimodal IP presented by [19] to incremental multimodal output allocation and show how it can help to avoid frequent self-corrections or output modifications from the system that are the result of dynamically changing input hypotheses. While the benefit of generating fewer self-corrections is not specific to incremental systems, but can be generalised to all interactive systems, we assume here that incremental systems face a particular danger of self-correcting too often due to their increased number of hypothesis updates.

[1] Heriot-Watt University, School of Mathematical and Computer Sciences, Edinburgh, Scotland, email: n.s.dethlefs@hw.ac.uk, v.t.rieser@hw.ac.uk, h.hastie@hw.ac.uk, o.lemon@hw.ac.uk

In Proceedings of the ECAI Workshop on Machine Learning for Interactive Systems: Bridging the Gap Between Language, Motor Control and Vision, Montpellier, France, pages 31-36, 2012.

As a domain of application, we address the information presentation phase in an interactive system for restaurant recommendations, extending previous work by [7], who present an incremental version of the work by [21]. While this previous work has focused on choosing a suitable presentation strategy for verbal presentation, here we focus on choosing the best modality accompanying a list of database hits. We assume that the choice of attributes (i.e. attributes that the user wishes the search to focus on) is determined by matching the types specified in the user input. Attributes include the *cuisine, food quality, location, price range* and *service quality* of a restaurant. The system then performs a database lookup and chooses a multimodal presentation strategy among *verbalOnly* and *combinedModalities*, i.e. visual and verbal output together. Visual output in this context refers to displays, on a screen or mobile device, that inform the user of the system's current best input hypotheses. Figure 1 shows examples of the main types of multimodal presentation strategies. The system does not have the option to present only visual information, since a Wizard-of-Oz study by [19] showed that human wizards never chose this strategy.

3 Optimising Multimodal Output Generation in Incremental Dialogue

3.1 Reinforcement Learning

To optimise the multimodal output generation process within an incremental model of dialogue processing, we define an RL agent as a Markov Decision Process, or MDP, which is characterised as a four-tuple $< S, A, T, R >$, where S is a set of states representing the status of the output generator and all information available to it; A is a set of output generation actions that combine strategies for multimodal IP with handling incremental updates in the system; T is a probabilistic transition function that determines the next state s' from the current state s and the action a according to a conditional probability distribution $P(s'|s, a)$; and R is a reward function that specifies the reward (a numeric value) that an agent receives for taking action a in state s.

Using such an MDP, the output generation process can be seen as a finite sequence of states, actions and rewards $\{s_0, a_0, r_1, s_1, a_1, ..., r_{t-1}, s_t\}$, where t is the time step. Every learning episode falls naturally into a number of time steps at each of which the agent observes the current state of the environment s_t, takes an action a_t and makes a transition to state s_{t+1}. This mechanism also defines the principle for the agent's micro-turn taking behaviour: it checks at each time step whether the state of the environment has changed so that an output action is required, e.g. if new input has come in or old input has been revised. If no particular action is required, e.g. because the user is still speaking, the agent may also decide to do nothing for the moment. Once information has been presented to the user, it is *committed* or *realised*. Here is where the difference between modalities may become most obvious to the user. While verbal output, once communicated to the user, cannot be changed without an explicit self-correction that marks the changed hypothesis, visual output can be changed more straightforwardly through an updated visual display, which may cause less disruption to an interaction.

The ultimate goal of an MDP is to find an optimal policy π^* according to which the agent receives the maximal possible reward for each visited state. We use the Q-Learning algorithm [29] to learn an optimal policy according to

$$\pi^*(s) = \arg\max_{a \in A} Q^*(s, a), \quad (1)$$

States
dataBaseHits {0=none,1=few,2=medium,3=many}
incrementalStatus {0=none,1=holdFloor,2=correct,3=selfCorrect}
modalityStatus {0=none,1=verbalOnly,2=combined}
statusCuisine {0=unfilled,1=low,2=medium,3=high,4=realised}
statusFood {0=unfilled,1=low,2=medium,3=high,4=realised}
statusLocation {0=unfilled,1=low,2=medium,3=high,4=realised}
statusPrice {0=unfilled,1=low,2=medium,3=high,4=realised}
statusService {0=unfilled,1=low,2=medium,3=high,4=realised}
userReaction {0=none,1=select,2=askMore,3=other}
userSilence={0=false,1=true}

Actions

Slot-ordering: presentCuisine, presentFood, presentLocation, presentPrice, presentService,

Incremental: backchannel, correct, selfCorrect, holdFloor, waitMore

Modality: verbalOnly, combinedModalities

Goal State ?, 0, $\geq$ 1, 0 $\vee$ 4, 0 $\vee$ 4, 0 $\vee$ 4, 0 $\vee$ 4, 0 $\vee$ 4, 1, 0 $\vee$ 1

Figure 2. The state and action space of the learning agent. The goal state is reached when all items (that the user may be interested in) have been presented and the most suitable output modality has been chosen. The goal state is defined with respect to the state variables above, where question marks indicate that the variable's value is irrelevant for reaching the goal state.

where Q^* specifies the expected reward for executing action a in state s and then following policy π^*.

3.2 The State and Action Space

The agent's state space needs to contain all information relevant for choosing an optimal strategy for multimodal output generation and an optimal sequence of incremental actions. Figure 2 shows the state and action space of our learning agent. The states contain information on the incremental, multimodal and attribute presentation status of the system.

The variable 'incrementalStatus' characterises situations in which a particular (incremental) action is triggered. For example, a `holdFloor` is generated when the user has finished speaking, but the system has not yet finished its database lookup. A `correction` is needed when the system has to modify already presented information (because the user changed their preferences) and a `selfCorrection` is needed when previously presented information is modified because the system made a mistake (in recognition or interpretation).

The variables representing the status of the cuisine, food, location, price and service indicate whether the slot is of interest to the user (0 means that the user does not care about it), and what input confidence score is currently associated with its value. Once slots have been presented, they are *realised* and can only be changed through a correction or self-correction.

The variable 'userReaction' shows the user's reaction to an IP episode. The user can select a restaurant, provide more information to further constrain the search or do something else. The 'userSilence' variable indicates whether the user is speaking or not. This can be relevant for holding the floor or generating backchannels.

The focus of this paper lies in the optimisation of multimodal output generation for incremental IP settings and is represented by the 'modalityStatus' variable and its accompanying action set of *verbalOnly* and *combinedModalities* (shown in bold-face fonts in Figure 2). The agent will learn to choose the best multimodal output genera-

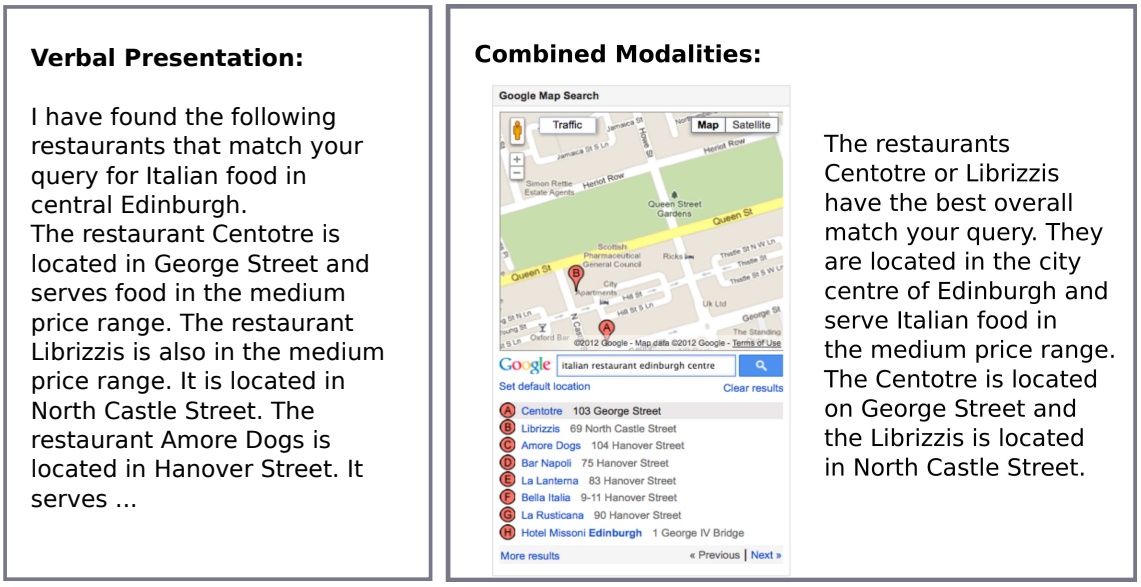


Figure 1. Examples of the different modalities we are considering for information presentation. The system can choose an exclusively *verbal presentation*, and verbalise all restaurant options it retrieved (left-hand side). Alternatively, the system can choose to *combine verbal and visual* output and present a map of the area along with a list of possible options and a verbalisation of those options that best match the user's query (right-hand side).

tion strategy based on the other available variables, in particular with respect to the (discretised) number of retrieved database hits and the agent's user input confidence scores. We do not consider a *visualOnly* presentation strategy in this paper, since this action was never chosen by human users in the Wizard-of-Oz data that underlies our training environment [20]. In future work, we aim to include such a presentation strategy and investigate its impact within our framework.

The complete state-action space size of this agent is roughly 10 million. The agent reaches its goal state (defined w.r.t. the state variables in Figure 2) when a multimodal output IP strategy has been chosen and all relevant attributes have been presented.

3.3 The Simulated Environment

We train our learning agent in a simulated environment with two components, one for estimating user reactions to multimodal IP strategies and one for simulating dynamically updated input hypotheses within the incremental dialogue setting.

The first component deals with estimating user reactions to a multimodal information presentation strategy which contains the options *verbalOnly* and *combinedModalities*. This simulation component was trained from data (using the simulation described in [20]) and represents user reactions as bi-grams of the form $P(a_{u,t}|IP_{s,t})$, where $a_{u,t}$ is the predicted user reaction at time t to the system's IP strategy $IP_{s,t}$ in state s at time t. We distinguish the user reactions of *select* a restaurant, *addMoreInfo* to the current query to constrain the search and *other*.

While the multimodal IP strategies can be used for incremental and non-incremental output generation, the second part of the simulation deals explicitly with the dynamic environment updates during an interaction. We assume that for each restaurant recommendation, the user has the option of filling any or all of the attributes *cuisine, food quality, location, price range* and *service quality*. The possible values of each attribute and possible confidence scores are shown in Table 1 and denote the same as described in Section 3.2.

At the beginning of a learning episode, we assign each attribute a possible value and confidence score with equal probability. For food and service quality, we assume that the user is never interested in bad food or service. Subsequently, confidence scores can change at each time step. (In future work these transition probabilities will be estimated from a data collection, though the following assumptions are realistic, based on our experience.) We assume that a confidence score of 0 changes to any other value with a likelihood of 0.05. A confidence score of 1 changes with a probability of 0.3, a confidence score of 2 with a probability of 0.1 and a confidence score of 3 with a probability of 0.03. The new states that the agent makes a transition into are uniformly distributed. Once slots have been realised, their value is set to 4. Verbally presented slots cannot be changed then without an explicitly verbalised self-correction. We assume that realised slots change with a probability of 0.1. If they change, we assume that half of the time, the user is the origin of the change (because they changed their mind) and half of the time the system is the origin of the change (because of an ASR or interpretation error). Each time a confidence score is changed, it has a probability of 0.5 to also change its value. The resulting input to the NLG component are data structures of the form *present(cuisine=Indian), confidence=low*.

Attribute	Values	Confidence
Cuisine	Chinese, French, German, Indian, Italian, Japanese, Mexican, Scottish, Spanish, Thai	0, 1, 2, 3, 4
Food	bad, adequate, good, very good	0, 1, 2, 3, 4
Location	7 distinct areas of the city	0, 1, 2, 3, 4
Price	cheap, expensive, good-price-for-value, very expensive	0, 1, 2, 3, 4
Service	bad, adequate, good, very good	0, 1, 2, 3, 4

Table 1. User goal slots for restaurant queries with possible values and confidence scores.

3.4 The Reward Function

The main trade-off that the learning agent needs to optimise is to find the best multimodal information presentation strategy given the number of database hits for the user's query and the confidence scores held for attributes that represent the user's preferences. To learn an action policy for this problem, we use the reward function suggested by [20], which was induced from human data using a multiple linear regression analysis. It aims to optimise *task ease*, which is a combined value of the metrics *The task was easy to solve* and *I had no problems finding the information I wanted*. Human users had originally assigned scores to these metrics in a Wizard-of-Oz study. [2] The reward function is defined as follows.

$$\mathrm{R} = \begin{cases} -20.2 & \times \text{dialogueLength} + \\ 11.8 & \times \text{taskCompletion} + \\ 8.7 & \times \text{multimodalScore} . \end{cases} \quad (2)$$

The value for *dialogueLength* here corresponds to the number of dialogue turns until the user has selected a restaurant. The value for *taskCompletion* is a discretised score indicating whether the system has been able to successfully make a restaurant recommendation. It is $+10$ if the user selects a restaurant and -10 otherwise. The value *multimodalScore*, finally, indicates the appropriateness of the chosen presentation strategy estimated from human behaviour in a Wizard-of-Oz study, please see [20] for details. The score is related to the number of database hits presented using each modality through curve fitting. This technique selects the most likely model for the data based on function interpolation. In terms of rewards for a multimodal (or combined) output, it yields a quadratic function that assigns a maximal score to a strategy displaying 14.8 items. This number corresponds to the curve inflection point. For an exclusively verbal presentation, the reward is computed based on a linear function which assigns negative scores to all presented items ≥ 4.

Rewards according to Equation 2 are assigned at the end of an episode, which stretches from the moment that a user specified their initial restaurant preferences to the moment in which they choose a restaurant (or reject all presented choices). In addition, we assign a number of rewards during the course of an episode that are directed at the incremental dialogue setting. The agent receives a reward of 0 whenever the user adds more information to the query, a reward of -10 for generating a (verbal or partially verbal) self-correction, -0.5 for holding the floor and an increasing negative reward for waiting $waiting\_time^2$ (to the power of two), in terms of the number of time steps passed since the last item was presented. This reward is theoretically $-\infty$ so that the agent is penalised stronger the longer it delays to begin the information presentation phase. Using this reward function, the agent was trained for 10 thousand learning episodes.

4 Experimental Results

After training, the agent has learnt the following strategy for multimodal output generation in an incremental dialogue setting. It will choose an exclusively verbal presentation strategy whenever the search has returned few items (up to four) and the confidence in their values is relatively high (or at least medium). For a medium number of items to present (i.e. more than four but less than 30), the agent will choose a combined strategy of verbal and visual output if its confidence in the requested attributes is relatively high. If its confidence is low, it will first only display visual information and delay the verbal presentation as long as possible, waiting for confidence scores to stabilise. The same is true for a large number of items to present. In other words, the agent learns to prefer to include visual information whenever it is not confident (enough) of its current user input hypotheses. In this way, it is able to increase its dialogue efficiency because users are given a chance to restate their preferences when they realise (through a visual display of the system's input hypotheses) that the system is currently working with a wrong input hypothesis. The agent is also able to reduce the number of its own verbal self-corrections (because visual displays can be updated without the need for an explicit correction). Note that due to our incremental setting, the multimodal presentation will typically precede the verbal presentation in order not to interrupt the user while they are still speaking. The system will thus present visual displays representing its current best hypothesis of the user's input and then, once the user has finished speaking, present the retrieved restaurant items verbally.

We designed three baselines to compare our approach with. The first baseline chooses among output modalities randomly, we call this baseline *RandomBase*. This baseline was designed to test whether modality allocation has an impact on task ease, at all. The second baseline was designed to compare our multimodal approach with a system that presents information only verbally. This baseline was used to test whether the visual information that is displayed during processing to inform the user about the system's current hypotheses was indeed helpful to increase task ease and reduce the number of dialogue turns and system self-corrections. We call this baseline *VerbalBase*. Finally, we designed a third baseline which always presents information combing verbal and visual information. We call this baseline *combinedBase*. This baseline tests the added value of incremental modality allocation. Note that all systems, including the baselines, learn to optimise the order of information presentation (as described in [7]) and therefore have a learning curve.

Figure 3 shows the learning curves for the learnt policy and the baselines and compares them according to their average reward (averaged over ten sample runs). The average reward attained by each policy defines their degree of *task ease* as specified in the reward function. As expected, *RandomBase* performs worst and is outperformed by the learnt policy by 44.8% ($p < 0.0001$, according to a t-test). The low performance of this baseline is likely due to its multimodal allocation actions not being sensitive to the number of retrieved database hits nor to the agent's current confidence scores of incoming user input. While the other two baselines also show non-optimal behaviour, their action policies are at least consistent, which in the long run gives them a higher chance of choosing an appropriate modality 'by chance'.

VerbalBase, which presents all information verbally, performs 15.2% worse than the learnt policy ($p < 0.0001$). Again, this baseline fails to take the number of retrieved database hits into account. What is worse, though, is that the policy at times starts presenting results when it is still not confident enough in the user's preferred values. It may thus start to present wrong information to the user and eventually be forced to self-correct, which incurs a high negative reward. While the system has the option to delay the information presentation phase as much as possible by choosing to *waitMore*, the waiting action also incurs an increasing negative reward which eventually forces the agent to start its verbal presentation.

CombinedBase, which always combines multimodal and verbal output, finally performs only 9.9% worse than the learnt policy

[2] Note that even though our setting is not identical to the one used by [20], we assume that the reward function is to an extent transferable to our domain, which is also a slot-filling application with relatively short episodes. In the future, we aim to learn a separate reward function that is specifically tailored towards our incremental setting.

($p < 0.0001$) and is therefore the best performing baseline. The reason is that this baseline is only affected by a non-optimal multimodal allocation, but significantly less by the problem of low confidence in user input hypotheses. The combined modality policy has the option of holding back the verbal presentation until it is confident in its input hypotheses, and is free to modify its visual presentation as much as possible, since a visual display does not need to be self-corrected verbally (and thus does not incur the negative reward associated with a verbal self-correction). [3] The primary source of negative rewards in this setting is therefore the suboptimal multimodal strategy chosen when compared to the human strategies preferred in the Wizard-of-Oz study, based on which we trained our simulation and reward function.

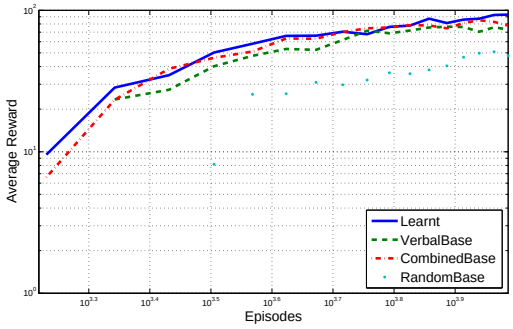


Figure 3. Learning curves indicating the average rewards, i.e. the average degree of task ease, attained by each policy.

5 Discussion

This paper has presented a preliminary investigation into how multimodal output generation can be integrated into incremental dialogue systems that process user input and plan system output in a parallel fashion. Our considerations here have been guided by how the *task ease* of a possible restaurant recommendation application for mobile devices can be optimised, in particular by increasing dialogue efficiency by multimodal display and reducing the number of verbal self-corrections that are caused by dynamically changing user input hypotheses. As is the nature of a proof-of-concept study, results are preliminary and so far based on simulation only. A variety of extensions of this work are possible. Importantly, we have not considered the restrictions that properties of the physical situation, the user or the particular application may pose on the choice of output modality, In in-car applications, for example, if we have indication of a high cognitive load or stress level (e.g. the eyes are fixed on the street) as in [9], the system could delay the presentation until a more suitable situation arrives and, simultaneously, mark the delay by a hesitation signal such as a turn holder. Similarly, we have left the question of user input modalities unaddressed and assumed that users always provide speech input.

The physical location of the user can have an impact on the preferred output modality in several ways. In crowded places, for instance, the system (and the user) may prefer a multimodal display due to the noise conditions that are likely to affect ASR results. Similarly, the system may take the user's current GPS position into account for its database lookup and prefer restaurants that are located close to the user's current location.

In terms of restrictions posed by the user, it is well known that individual users differ with respect to their specific preferences with regard to semantic [25] and lexical-syntactic [26] choices in language production. There is thus reason to expect that individual users will also have preferences for certain output modalities, some preferring verbal presentations, some visual output and combinations of different sorts. As a system 'gets to know' its user better, it may therefore want to increasingly take its particular user's preferences into account when choosing an output modality.

In addition, certain applications may themselves restrict the possible input and output modalities that a system can rely on. Many *hands-free* and *eyes-free* scenarios, such as an in-car mobile device, require the user to use speech only, or buttons that are manufactured into the steering wheel, to specify their search queries, and at the same time, should not be followed by multimodal output of the system that may require the driver to take their eyes off the traffic. On the other hand, previous work has shown that noisy ASR can distract drivers just as much [14], so that finding an appropriate multimodal output combination could amount to a challenging task.

There is also no obvious reason to restrict the user's input modalities to speech only. Instead, previous work has shown that a combination of speech and text input can lead to more efficient interactions when users are allowed to (incrementally) modify their search queries and retrieved results [13]. This can lead to decreased mental demand, perceived effort and level of frustration.

Finally, we have not paid explicit attention to the synchronisation between the different modalities, but have rather assumed that since output modalities are decided at the micro-turn level, they will automatically synchronise at the level of the utterance. While for the present (simulation-based) study, this has not presented a problem, it needs to be determined whether in practice a more principled mechanism for synchronisation is needed. An interesting direction, for example, could be to insert location points of restaurants on a map gradually, as they are presented as speech output in parallel.

6 Conclusion and Future Directions

This paper has presented a proof-of-concept study for optimising multimodal output generation for information presentation for incremental dialogue systems, i.e. systems that perform processing of user input and planning of system output in a parallel fashion. In particular, we have used Reinforcement Learning to optimise the *multimodal allocation* of our system, that is, to find an optimal combination of modalities for every given context. Preliminary results based on a partially data-driven user simulation are promising. They indicate that the agent is able to optimise its modality allocation by choosing an exclusively verbal presentation strategy for few search results and relatively high confidence scores in user input hypotheses. Alternatively, the agent can choose a strategy that combines visual and verbal output for a higher number of search results or situations involving low confidence scores in user input hypotheses. In this way, the resulting dialogues have gained in *task ease*, which was suggested by significantly higher rewards, shorter dialogues and fewer self-corrections which our system produced in comparison to a number of hand-crafted baselines.

In future work, we would like to extend our suggested model and re-train it using a fully data-driven simulated environment and reward function based on a data collection that explicitly addresses incremental discourse phenomena. This would allow us to explicitly

[3] While we did not restrict the number of visual updates in this setting, in practice, such a restriction may be necessary in order not to confuse users.

take the real-time nature of our model into account and not only estimate how input confidence scores change over time, but also how user behaviour changes through the incremental nature of our dialogue framework.

Further possible directions include the use of multiple user input modalities, adaptation to individual users during an interaction using online learning and a comprehensive evaluation of our suggested method using human users in a real-world setting. A further possibility is a data collection in an incremental multimodal setting to learn more about the effects of combining incremental processing and multimodal output generation on human-computer interaction.

ACKNOWLEDGEMENTS

This research has received funding from EC's FP7 programmes: (FP7/2011-14) under grant agreement no. 287615 (PARLANCE); (FP7/2007-13) under grant agreement no. 216594 (CLASSiC); (FP7/2011-14) under grant agreement no. 270019 (SPACEBOOK); (FP7/2011-16) under grant agreement no. 269427 (STAC).

REFERENCES

[1] Elisabeth André and Thomas Rist, 'Presenting through performing: on the use of multiple lifelike characters in knowledge-based presentation systems', *Knowledge Based Systems*, **14**, 3–13, (2001).

[2] Timo Baumann, Okko Buss, and David Schlangen, 'Evaluation and Optimisation of Incremental Processors', *Dialogue and Discourse*, **2(1)**, (2011).

[3] Hendrik Buschmeier, Timo Baumann, Benjamin Dosch, Stefan Kopp, and David Schlangen, 'Combining Incremental Language Generation and Incremental Speech Synthesis for Adaptive Information Presentation', in *Proceedings of 13th Annual SIGdial Meeting on Discourse and Dialogue*, Seoul, South Korea, (2012).

[4] Okko Buss, Timo Baumann, and David Schlangen, 'Collaborating on Utterances with a Spoken Dialogue Systen Using an ISU-based Approach to Incremental Dialogue Management', in *Proceedings of 11th Annual SIGdial Meeting on Discourse and Dialogue*, (2010).

[5] Heriberto Cuayáhuitl and Nina Dethlefs, 'Hierarchical Multiagent Reinforcement Learning for Coordinating Verbal and Nonverbal Actions in Robots', in *Proceedings of the ECAI Workshop on Machine Learning for Interactive Systems: Bridging the Gap Between Language, Motor Control and Vision (MLIS-2012)*, (2012).

[6] Nina Dethlefs, Helen Hastie, Verena Rieser, and Oliver Lemon, 'Optimising Incremental Dialogue Decisions Using Information Density for Interactive Systems', in *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Jeju, South Korea, (2012).

[7] Nina Dethlefs, Helen Hastie, Verena Rieser, and Oliver Lemon, 'Optimising Incremental Generation for Spoken Dialogue Systems: Reducing the Need for Fillers', in *Proceedings of the International Conference on Natural Language Generation (INLG)*, Chicago, Illinois, USA, (2012).

[8] David DeVault, Kenji Sagae, and David Traum, 'Can I finish? Learning when to respond to incremental interpretation result in interactive dialogue', in *Proceedings of the 10th Annual SigDial Meeting on Discourse and Dialogue*, Queen Mary University, UK, (2009).

[9] M. Gasic, P. Tsiakoulis, M. Henderson, B. Thomson, K. Yu, E. Tzirkel, and S. Young, 'The effect of cognitive load on a statistical dialogue system', in *Proc. of SIGdial Workshop on Discourse and Dialogue*, (2012).

[10] Alexander Gruenstein, Stephanie Seneff, and Chao Wang, 'Scalable and Portable Web-Based Multimodal Dialogue Interaction with Geographical Databases', in *Proceedings of INTERSPEECH*, (2006).

[11] Hui Guo and Amanda Stent, 'Trainable Adaptable Multimedia Presentation Generation', in *Proceedings of the International Conference on Multimodal Interfaces (ICMI)*, (2005).

[12] Anne Kilger and Wolfgang Finkler, 'Incremental generation for real-time applications', Technical report, DFKI Saarbruecken, Germany, (1995).

[13] Anuj Kumar, Tim Paek, and Bongshin Lee, 'Voice Typing: A New Speech Interaction Model for Dictation on Touchscreen Devices', Austin, Texas, USA, (2012).

[14] Andrew Kun, Tim Paek, and Jeljko Medenica, 'The Effect of Speech Interface Accuracy on Driving Performance', in *Proceedings of INTERSPEECH*, (2007).

[15] Sharon Oviatt, Rachel Coulston, and Rebecca Lunsford, 'When do we interact multimodally? Cognitive load and multimodal communication patterns', in *Proceedings of the International Conference on Multimodal Interfaces (ICMI)*, (2004).

[16] Tim Paek, Bo Thiesson, and Y.C. Ju andBongshin Lee, 'Search Vox: Leveraging Multimodal Refinement and Partial Knowledge for Mobile Voice Search', in *Proceedings of User Interface Software and Technology (UIST)*, (2008).

[17] Matthew Purver and Masayuki Otsuka, 'Incremental Generation by Incremental Parsing', in *Proceedings of the 6th UK Special-Interesting Group for Computational Linguistics (CLUK) Colloquium*, (2003).

[18] Antoine Raux and Maxine Eskenazi, 'A Finite-State Turn-Taking Model for Spoken Dialog Systems', in *Proceedings of the 10th Conference of the North American Chapter of the Association for Computational Linguistics—Human Language Technologies (NAACL-HLT)*, Boulder, Colorado, (2009).

[19] Verena Rieser and Oliver Lemon, 'Learning Effective Multimodal Dialogue Strategies from Wizard-of-Oz Data: Bootstrapping and Evaluation', in *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics (ACL/HLT)*, (2008).

[20] Verena Rieser and Oliver Lemon, 'Learning and Evaluation of Dialogue Strategies for new Applications: Empirical Methods for Optimization from Small Data Sets', *Computational Linguistics*, **37(1)**, 153–196, (2011).

[21] Verena Rieser, Oliver Lemon, and Xingkun Liu, 'Optimising Information Presentation for Spoken Dialogue Systems', in *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics (ACL)*, Uppsala, Sweden, (2010).

[22] David Schlangen and Gabriel Skantze, 'A General, Abstract Model of Incremental Dialogue Processing', in *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics*, Athens, Greece, (2009).

[23] Ethan Selfridge, Iker Arizmendi, Peter Heeman, and Jason Williams, 'Stability and Accuracy in Incremental Speech Recognition', in *Proceedings of the 12th Annual SigDial Meeting on Discourse and Dialogue*, Portland, Oregon, (2011).

[24] Gabriel Skantze and Anna Hjalmarsson, 'Towards Incremental Speech Generation in Dialogue Systems', in *Proceedings of the 11th Annual SigDial Meeting on Discourse and Dialogue*, Tokyo, Japan, (2010).

[25] Jette Viethen and Robert Dale, 'The Use of Spatial Relations in Referring Expression Generation', in *Proceedings of the International Conference on Natural Language Generation (INLG)*, (2008).

[26] Marilyn Walker, Amanda Stent, François Mairesse, and Rashmi Prasad, 'Individual and Domain Adaptation in Sentene Planning for Dialogue', *Journal of Artificial Intelligence Research (JAIR)*, **30**, 413–456, (2007).

[27] Marilyn Walker, Steve Whittaker, Amanda Stent, Pretaam Maloor, Johanna Moore, and G Vasireddy, 'Generation and Evaluation of User Tailored Responses in Multimodal Dialogue', *Cognitive Science*, **28(5)**, 811–840, (2004).

[28] Kuansan Wang, 'A Study on Semantic Synchronous Understanding on Speech Interface Design', in *Proceedings of UIST-2033*, Vancouver, Canada, (2003).

[29] Chris Watkins, *Learning from Delayed Rewards*, PhD Thesis, King's College, Cambridge, UK, 1989.

Learning Hierarchical Prototypes of Motion Time Series for Interactive Systems

Ulf Großekathöfer[1] and **Shlomo Geva**[2] and **Thomas Hermann**[1] and **Stefan Kopp**[1]

Abstract. For interactive systems, recognition, reproduction, and generalization of observed motion data are crucial for successful interaction. In this paper, we present a novel method for analysis of motion data that we refer to as K-OMM-trees. K-OMM-trees combine Ordered Means Models (OMMs) a model-based machine learning approach for time series with an hierarchical analysis technique for very large data sets, the K-tree algorithm. The proposed K-OMM-trees enable unsupervised prototype extraction of motion time series data with hierarchical data representation. After introducing the algorithmic details, we apply the proposed method to a gesture data set that includes substantial inter-class variations. Results from our studies show that K-OMM-trees are able to substantially increase the recognition performance and to learn an inherent data hierarchy with meaningful gesture abstractions.

1 Introduction

Organizing complex (body) motion data is a challenging task in todays robots/virtual agents research. Even thought robots and agents are equipped with reliable sensing technology such as time-of-flight cameras, tactile sensor grids, etc., the ability to coordinate and categorize observed motion data is still far behind. When humans perform meaningful body motions such as arm gestures, human observers are easily able to incrementally and abstractly structure arising gestures by common characteristics as orientation, shape, size, velocity etc. In addition, humans reproduce these gestures with their own body according to the learned structure. Furthermore, a human observer is capable to rapidly generalize from these characteristics and recognize a gesture's abstract meaning independently from its orientation, size, velocity, shape etc. This is, for example, the case in humans that communicate by means of sign language signs: the interactants are able to recognize, structure, and reproduce sign gestures, whereby they make sense of their observations in an incremental way, eventhough sign language signs cover substantial inter-personal variations.

Thus, in order to benefit from advanced sensing technologies, computer systems, robots, or agents are confronted with various technical challenges: (1.) data organization, i.e., in what way the observations are structured and how motion abstractions are represented, (2.) rapid, on-line and adaptive processing of observed motions, i.e., to incrementally update the data model if new observations arrive, (3.) reproduction of motion data. i.e., to allow computer systems to learn body motion and motion abstractions by imitation. In this paper, we understand body motions as multivariate time series of sensor values. For analysis of those sensor values, we present K-OMM-trees – a machine learning approach that is able to (1.) extract meaningful prototypes of body gesture data in an unsupervised manner, (2.) hierarchically represent these prototype gestures, and (3.) in principal, incrementally update the learned structure as new observations arrive. We apply K-OMM-trees to data sets related to gesture recognition. After discussing existing approaches to learning of body motions in Section 2, we introduce the K-OMM-trees algorithm in Section 3. Then, we present our gesture study and discuss the results in Section 4, followed by a conclusion in Section 5.

[1] Cognitive Interaction Technology Center of Excellence (CITEC), Bielefeld University, Germany, email: ugrossek@techfak.uni-bielefeld.de

[2] Queensland University of Technology, Brisbane, Australia, email: s.geva@qut.edu.au

2 Related Work

In this paper, we represent body motion data as multivariate time series, i.e., as multi-dimensional variables that vary in time, for example, as location and orientation coordinates of body parts such as hand wrists, and/or additional features such as joint angle, pen pressure, etc. Representing gestures and body motions as multivariate time series induces various technical demands, and, therefore, algorithms have to consider the particular properties of this data type. Challenges in analysis of time series concern observations of variable lengths and non-linear compressions or expansions of the time axis; the application of standard machine learning algorithms to time series data can be difficult and likely requires a data pre-processing that might induce information losses. As a consequence, the analysis of time series requires specialized techniques such as dynamic time warping (DTW, [3]).

A common approach for representation and generation of motion data are Hidden Markov Models (HMMs) [13, 14, 17, 20]. HMMs have been applied to representation of body motions, e.g., Calinon et al. [2] used HMMs together with Gaussian mixture regression to generalize motion demonstrations during reproduction. Inamura et al. [11] applied HMMs to both recognition and generation of full body behavior; Amit and Mataric [1] implemented gesture imitation learning based on preprocessed and abstracted time series of joint values. Ordered Means Models (OMMs) as an easy-to-use approach for gesture data recognition and generation have been introduced lately [8, 9, 10, 21]. OMMs are based on a simple model ar-

In Proceedings of the ECAI Workshop on Machine Learning for Interactive Systems: Bridging the Gap Between Language, Motor Control and Vision, Montpellier, France, pages 37-42, 2012.

chitecture that allows a computationally efficient training of time series data. In addition, OMMs are able to capture and explicitly represent relevant characteristics of an underlying time series distribution, and their model parameters can be fully analyzed in terms of a prototype representation in the original motion space [9].

3 K-OMM-trees

In this paper, we use a novel algorithmic approach to unsupervised organization of body motion data, which we refer to as K-OMM-trees. K-OMM-trees are hybrids of OMMs and K-trees. After introducing the OMM-algorithm, we will outline a clustering approach for motion time series (*K-OMMs*), and, finally, present the K-OMM-tree algorithm.

3.1 Ordered Means Models

An ordered means model (OMM) is a generative, discrete state space model that emits a time series out of an adjustable number K of model states. In an OMM, only transitions to states with equal or higher indexes as compared to the current state are allowed, i.e., the network of model states and state transitions follows a left-to-right topology.

Let $O = \mathbf{o}_1 \dots \mathbf{o}_T$ be a (multivariate) time series of observation vectors with $\mathbf{o}_t \in \mathbb{R}^d$. Given the above model structure, an OMM is defined by a set of emission densities $B = \{b_k(\mathbf{o_t})\}$, each of which is associated with a particular state. To model the emission densities we use Gaussian densities $b_k(\mathbf{o_t}) = g(\mathbf{o_t}; \boldsymbol{\mu}_k, \sigma)$, where σ is identical in all states and is used as a global hyper-parameter.

The global standard deviation and the left-to-right model topology gives rise to models that are mainly defined by a linear array of reference vectors $[\boldsymbol{\mu}_1 \dots \boldsymbol{\mu}_K]$, i.e., the location parameters of the emission densities.

In principle, OMMs require the definition of an explicit length distribution $P(T)$ either by domain knowledge or by estimation from the observed lengths in the training data which may not be possible due to missing knowledge or non-representative lengths of the observations. We assume a flat distribution in terms of an improper prior according to equally probable lengths. For a given length T, we define each valid path $\mathbf{q}_T = q_1 \dots q_T$ through the model to be equally likely:

$$P(\mathbf{q}_T|\Omega) = \begin{cases} \frac{1}{M_T} \cdot P(T) & \text{if } q_1 \le q_2 \le .. \le q_T, \\ 0 & \text{else} \end{cases} \quad (1)$$

where M_T is the number of valid paths for a time series of length T through a K-state model:

$$M_T = |\{\mathbf{q}_T : q_1 \le q_2 \le \dots \le q_T\}| = \binom{K+T-1}{T}. \quad (2)$$

The likelihood to observe the time series O for a given path $\mathbf{q}_T$ and a model Ω then is

$$p(O|\mathbf{q}_T, \Omega) = \prod_{t=1}^{T} p(\mathbf{o_t}|q_t, \Omega) = \prod_{t=1}^{T} b_{q_t}(\mathbf{o_t}). \quad (3)$$

The likelihood to observe the time series O and a path $\mathbf{q}_T$ for a given model Ω is

$$p(O, \mathbf{q}_T|\Omega) = p(O|\mathbf{q}_T, \Omega) \cdot P(\mathbf{q}_T|\Omega) \quad (4)$$

and the overall production likelihood for a time series of length T corresponds to the sum of all possible paths in Eq. 4

$$p(O|\Omega) = \sum_{\mathbf{q}_T} p(O, \mathbf{q}_T|\Omega). \quad (5)$$

3.1.1 Differences to HMMs

A well-known limitation of HMMs is that the state duration probabilities are geometrically distributed, which might not be compatible to a particular distribution found in a set of observations. For example, the temporal characteristics of speech are not well represented by a geometric distribution as modeled by standard HMM approaches [12, 15, 17]. To circumvent this restriction, various approaches have been proposed that incorporate the state duration into the model definition. E.g., Hidden Semi-Markov Models such as Explicit Duration HMMs (EHMMs) introduced by Ferguson [6] and the Continuously Variable Duration HMMs (CVDHMMs) from Levinson [15], replace the transition probabilities by an explicit state duration distributions. Other approaches, such as Variable Transition HMMs (VTHMMs, e.g. [19]), model the transition probabilities as a function of the elapsed duration, while Expanded State HMMs (ESHMMs) define sub-HMMs in each state that share the same emission densities. However, none of these approaches reduces the overall complexity of Hidden Markov Models.

In contrast, the radically simplified model design of OMMs leads to implicitly modeled duration probabilities. Here, the probability $P_k(\tau)$ to stay τ time steps in state k depends on the sequence length T and the number of model states K. Considering the combinatorics of the path generation process (see Eq. 1 and Eq. 2), the duration probability distributions of OMMs follow

$$P_k(\tau) = \frac{\binom{T+K-2-\tau}{K-2}}{\binom{T+K-1}{K-1}}, \quad (6)$$

with an expected value of the state duration of $\frac{T}{K}$.

3.1.2 EM-Training

To learn the model parameters $\boldsymbol{\mu}_k$ from a set of N example sequences $\mathbf{O} = \{O^1, \dots, O^N\}$, we maximize the log-likelihood

$$\mathcal{L} = \sum_{i=1}^{N} \ln p(O^i|\Omega) \quad (7)$$

with respect to the mean vectors $\boldsymbol{\mu}_k$.

To solve this optimization problem, we use an iterative approach that is similar to Baum-Welch training of HMMs. The optimization implies a training scheme based on the expectation maximization algorithm (cf. [5]) where we first compute the *responsibilities*

$$r^i_{k,t} = \frac{p(O^i, q_t = k|\Omega)}{p(O^i|\Omega)} \quad \text{(E-step)} \quad (8)$$

and then re-estimate the model parameters according to

$$\boldsymbol{\mu}_k = \frac{\sum_{i=1}^{N}\sum_{t=1}^{T} r^i_{k,t} \cdot \mathbf{o}^i_t}{\sum_{i=1}^{N}\sum_{t=1}^{T} r^i_{k,t}} \quad \text{(M-step)}. \quad (9)$$

These two steps are repeated until convergence.

Figure 1. Here, six randomly selected gestures from the used data set are shown. The 3-dimensional trajectories are mapped to the x/y-axis; the color indicates the time during execution (from "blue" to "red").

3.1.3 Efficient Computation of Production Likelihoods and Responsibilities

To compute the production likelihood (Eq. 5) and the responsibilities (Eq. 8) in a computationally efficient way, we use a dynamic program similar to the forward-backward algorithm known from HMMs [17]. Therefore, we define the forward variable according to

$$\alpha_{k,t} = \alpha_{k,t-1} \cdot b_k(\mathbf{o}_t) + \alpha_{k-1,t} \propto p(\mathbf{o}_1..\mathbf{o}_t | q_t \leq k, \Omega). \quad (10)$$

Note that the current $\alpha_{k,t}$ depends only on the variable values of the previous state $k-1$ and of the previous point in time $t-1$. This yields an elegant and fast dynamic programming solution:

$$\alpha_{k,t} = \alpha_{k,t-1} \cdot b_k(\mathbf{o}_t) + \alpha_{k-1,t} \quad (11)$$

that is initialized with $\alpha_{k,0} = 1$, and $\alpha_{0,t} = 0$. In a similar way we can compute the backward variable

$$\beta_{k,t} = \beta_{k,t+1} \cdot b_k(\mathbf{o}_t) + \beta_{k+1,t} \propto p(\mathbf{o}_t..\mathbf{o}_T | q_t \geq k, \Omega) \quad (12)$$

by means of recursion, initialized with $\beta_{k,T+1} = 1$ and $\beta_{K+1,t} = 0$. Note that the terminal values for the forward and backward variables are equal and proportional to the production likelihood:

$$\alpha_{K,T} = \beta_{1,1} \propto p(O|\Omega). \quad (13)$$

By combination of equation 10 and 12 the responsibilities can be computed by

$$r_{k,t} = \frac{\alpha_{k,t-1} \cdot b_k(\mathbf{o}_t) \cdot \beta_{k,t+1}}{\alpha_{K,T}}. \quad (14)$$

3.2 Identification of Time Series Prototypes

An OMM Ω is completely represented by a linear array of reference vectors $\Omega = [\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_K]$, which corresponds to the location parameter of the emission distributions. Since the location parameter are elements of the same data space as the observed data examples, the series of reference vectors is fully interpretable as a time series prototype. This affords the reproduction ability, by simply traversing the model in left-to-right order. Please see our paper [9] for further details and demonstrations.

To use OMMs for unsupervised identification of a set of time series prototypes, we use (1) an adjusted K-means approach, and (2) a hierarchical clustering technique for time series that we refer to as K-OMM-trees.

3.2.1 K-OMMs

To discover a set $\{\Omega^1, .., \Omega^K\}$ of such time series prototypes in an unsupervised manner, we adapted a K-means procedure, in which we replace the cluster centroids by OMM prototypes and apply the production likelihood $p(O|\Omega)$ as a distance function.

K-means clustering is a commonly used method to partition a set of observations into K groups [16] that, ideally, are associated by a common quality. The general idea of K-means is to minimize an objective function that accumulates the distances between each observation O^i and a prototype C^j (often called "cluster centroid" or "cluster means") to which it is assigned to. If the observations are (multi-variate) time series, then specific time series distance functions and centroid representations are necessary. To apply OMMs to clustering analysis, a natural distance function between a sequence O^i and a model Ω^j is defined by the negative production log-likelihood

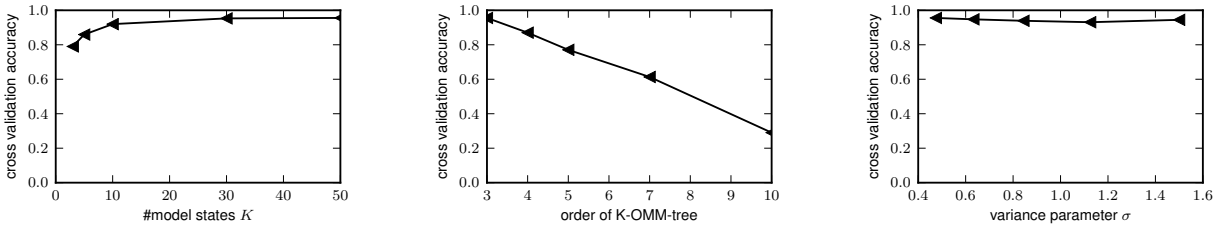
$$d(O^i, \Omega^j) = -\ln p(O^i|\Omega^j). \quad (15)$$

By this, the objective function of K-OMMs accumulates the likelihoods that the sequence O^i has been generated by the model Ω^j, which is used as the centroid. Again, the discovered K prototypes that function as centroids can be interpreted as time series prototypes for the set of associated time series. The update function in such an approach is a re-training of the models according to the assigned time series. Note that in case of K-means the number of prototypes K has to be determined by an external process, e.g., the elbow method.

3.2.2 K-OMM-trees

To enable hierarchical identification of time series prototypes, we use K-OMM-trees—an approach of which the structural properties are similar to those of K-trees [4, 7]. Generally, K-trees are height balanced clustering trees that combine classical K-means and B+-trees algorithms to highly time-efficient clustering trees. A K-tree of order m is defined as (cf. [7]):

1. All leaves are on the same level.
2. All internal nodes, including the root, have at most m nonempty children, and at least one nonempty child.
3. Clusters centroids act as search keys.
4. The number of keys in each internal node is equal to the number of its nonempty children, and these keys partition the keys in the children to form a nearest neighbor search tree.
5. Leaf nodes contain observations.



(a) K (b) *order* (c) σ

Figure 2. This figure shows the cross validation accuracy of for varying number of model states K (a), varying order of the K-OMM-tree (b), as well as varying variance parameter σ (c).

The time complexity of building K-tree is $O(n \log n)$.

To enable hierarchical analysis of time series by means of K-OMM-trees, we use a hybrid of K-OMMs clustering and K-trees in which the nodes of the tree are represented by OMM prototypes and the leafs hold the time series examples. The process of building a K-OMM-tree is similar to the building of a K-tree. However, in contrast to euclidean distances, we use the negative log-likelihood (cf. Eq. 15) as a distance function and re-train the OMM prototypes if a new time series is added to a branch. Since OMM are represented by an ordered array of reference vectors and are, therefore, time series themselves, it is possible to build "OMMs of OMMs", i.e., to train a model for a set of models. This unique property enables a fast training of K-OMM-trees: for nodes that contain sub-trees it is possible to build an OMM representation out of the OMM representations from the level below.

Building a K-OMM-tree is a dynamic process that inserts time series on-line, as they are observed. Consider an already existing K-OMM-tree of order m to which an additional time series O is presented for insertion. The insertion procedure is: Firstly, the tree is searched to identify the node that contains the time series nearest neighbor. If this node stores less than m —the tree's order— observations, O is inserted as a leaf. Since this insertion changes the cluster structure, the parent nodes up to the root node must update their search keys in terms or re-computing their OMMs. When a leaf stores $m+1$ time series, it can not contain any more observations. It then is split by applying the K-OMMs-algorithm with two clusters to the $m+1$ elements of that node. The resulting OMMs become the search keys for two new child nodes, each containing the associated vectors. Now consider that the parent of this new nodes contains $m+1$ search keys after the splitting process, too. Then it also has to be split, replacing itself in its parent by the two new child nodes. This process is repeated in a similar manner until a node with less than m search keys is reached, or the root node is replaced by a new root node. The procedure of building a K-OMM-tree is initialized with an empty root node and just one leaf.

3.3 Classification with OMMs and K-OMM-trees

To use OMMs for classification, i.e., to assign an unseen gesture to one of J classes, J class-specific models are firstly estimated from the data. Assuming equal prior probabilities, an unknown gesture O is assigned to the class associated with the model that yields the highest production likelihood $p(O|\Omega)$. In case of K-OMM-trees, all models Ω that are related to the tree and represent the tree's nodes, are used for classification.

4 Experiments

To demonstrate the capabilities of our algorithm, we designed an application setup that is optimized towards imitation learning during human-agent interaction. The scene comprises a virtual agent that learns gesture classes by means of a human demonstrator. In this study, we investigated the following research questions:

1. Does the unsupervised learned structure of K-OMM-trees support recognition of unseen gestures?
2. What influence do the chosen K-OMM-treehyper-parameters have on the performance?
3. Do the learned OMM prototypes of K-OMM-trees provide a model for gesture reproduction?

The setup comprises a time-of-flight camera, a marker-free tracking software and a humanoid virtual agent called *Vince*. The time-of-flight camera (a SwissRanger$^{\text{TM}}$ SR4000[3]) captures the scene in 3d at a frequency of ≈ 30 fps. The scene data are used by the software iisu$^{\text{TM}}$ 2.0[4] to map a human skeleton on the present user in the scene. We extract the relevant information of the skeleton, such as the user's height, spatial positions of the wrists and the center of mass to compute the normalized 3d positions of the wrists with respect to the user's body size. Within a body-correspondence-solver submodule, the wrists' positions are transformed (rotated and scaled) from the coordinate system of the camera to egocentric space of the virtual agent which stays face-to-face to the human demonstrator. In the current study we focus on the right wrist and record these data as time series for each performed gesture. During data acquisition, Vince imitates the subject's right hand movements in real time. In this way, the demonstrator receives visual feedback on how Vince would perform those gestures. It is worth noting that the ambiguous position of the elbow at each time step is not captured but computed with the aid of inverse kinematic [18].

Overall, 520 examples were captured in the format of 3d wrist movement trajectories with time stamps. Each trajectory starts from and ends at the rest position of the right

[3] http://www.mesa-imaging.ch
[4] http://www.softkinetic.net

(a) Root (b) Node level 1 (c) Node level 2 (d) Node level 3 (e) Node level 4 (f) Leaf: gesture

Figure 3. These plots illustrate the OMM prototypes that correspond to one branch of the learn K-OMM-tree for the class "circle". The plots are related to child-nodes from left to right, i.e., the left-most plot is on the tree's root level, the second plot on level 1 and so forth. The right plot shows a gesture performance in a leaf. Please note, that we used all 3 location coordinates in the training process; these figures only plot the x/y-coordinates. The trajectories are mapped on the x/y-axis; the color indicates the development in time (from "blue" to "red").

hand, whereas the gestures were demonstrated at different velocities and require an average execution of 4.75 seconds. The performed gestures are collected in nine abstract classes that range from conventional communicative gestures ("waving", "come here" and "calm down") over iconic gestures ("circle", "spiky", "square", "surface" and "triangle") to deictic gestures ("pointing"). These gestures have been performed with respect to some of the following variant features: size (e.g. small and big circle), performing space (e.g. drawing a circle at the right side or in front of oneself), direction (clockwise or counter-clockwise), orientation (horizontal or vertical), repetition (repeating some subparts of the movement, such as drawing a circle once or twice, or swinging the hand for several times during waving). Figure 1 illustrates six randomly selected gestures from the "circle" class. The gestures in the plots are mapped to the x/y coordinates.

We divided the data set into a training and a test set. Thereby, the data set that is used for training contains approximately two-thirds of all examples (362 gestures). The remaining 147 gesture performances were used for testing.

We used K-OMM-trees for classification of these abstract gesture classes. For comparison, we also tested the data set by means of an OMM- and an HMM-based classifiers. The HMM incorporates similar properties as the used OMMs, i.e., left-to-right topology and a global standard deviation parameter σ that is used as a hyper-parameter. We applied an identical procedure to all classifiers. Firstly, we identified optimal hyper-parameters by means of 5-fold cross validation We chose equal values for the hyper-parameters: the number of model states was $K \in \{3, 5, 10, 20, 50\}$; the set of values for the standard deviation parameter was $sigma \in \{0.5, 1.0, 1.5, 2.0\}$. In case of K-OMM-trees, we additionally chose five values for the tree's order $m \in \{3, 4, 5, 7, 10\}$. Subsequently, we trained a classifier using all training data and the hyper-parameter combination that is associated with the lowest cross validation error. To obtain the final error rate, we applied the resulting classifiers to the dedicated test data set.

In addition, we examined the resulting OMM parameters that represent the tree's nodes, as well as the emerging tree structure.

4.1 Results and Discussion

Table 1 displays the results from the classification experiments. In terms of cross validation and test set accuracy, K-OMM-trees classifiers clearly outperform OMM- and HMM-based classifiers by up to $\approx 22\%$. While K-OMM-trees achive a test set accuracy value of ≈ 0.91, OMM reach ≈ 0.74 and HMMs ≈ 0.81. Similarly, the cross validation accuracy of K-OMM-trees is with ≈ 0.95 substantially higher than the cross validation accuracy of OMM classifiers (≈ 0.73) and HMM classifiers (≈ 0.76).

Method	K	σ	order	cv accuracy	test set accuracy
K-OMM-trees	50	0.47	3	0.95	0.91
HMMs	51	0.47	-	0.76	0.81
OMMs	50	0.47	-	0.73	0.74

Table 1. This table denotes the results from the classification experiments of the arm-gesture data set for all evaluated methods. In detail, the hyper-parameters (number of model states K, the variance parameter σ, and the order of the K-tree m) as well as reached cross validation accuracy and test set accuracy are shown.

These results indicate that the hierarchical structure of K-OMM-trees induce a superior data representation, at least if, as given with the evaluated gesture data, the data set comprises a wide inter-class variability.

Figure 2 illustrates the influence of the chosen hyper-parameters for K-OMM-trees on the achived cross validation accuracy. Each plot displays the dependency of one varying hyper-parameter while both remaining parameters are fixed.

Here, only a varying order m leads to substantially varying cross validation accuracy. The K-OMM-tree classifier yields an accuracy of ≈ 0.29 with an order $m = 10$. In contrast, changes of the variance parameter σ are almost without consequences: here, $\sigma = 1.125$ yields the lowest accuracy of ≈ 0.93 – only ≈ 0.02 lower than the highest accuracy achived by a classifier trained with $\sigma = 0.47$. Lastly, variations in the number of model states K induces small changes in cross validation accuracy; in particular small K values lead to decreased classification performances, e.g., a K-OMM-tree classifier with $K = 3$ model states yields an accuracy value of ≈ 0.79. The fact that an increased tree's order m leads to considerable decreases in accuracy might originate in a well-suited representation of the data with lower order. This is very likely different for other data domains.

Figure 3 shows graphical representations for one branch of the tree associated to class "circle". The branch ranges from the root node (sub-figure (a)) to an actual gesture in a leaf (sub-figure (f)), i.e., the first 5 plots show OMM proto-

types that represent the tree's nodes and the last plot illustrates an "circle" gesture that was used for training of the K-OMM-tree.

These plots suggests that K-OMM-trees are able to hierarchically organize gesture trajectories whereby higher levels are associated to more abstract gestures and lower nodes are connected to more specific gesture performances. Additionally, the plots indicate that the learned hierarchical representation reflects an inherent data taxonomy. I.e., the models in figure 3 are related to gestures that are performed clockwise, another branch of the tree contains counter-clockwise circles, etc.

5 Conclusion

In this paper we have presented an approach for learning and organizing gesture trajectories. Using K-OMM-trees our approach is able to learn how to organize gestures hierarchically and to use this hierarchy for recognition and reproduction of gestures. The results from our classification experiments show that K-OMM-trees are able to recognize gesture data with substantial higher accuracy than traditional classifiers. We are thus poised to believe that the combination of improved accuracy and hierarchical data representations that yield direct interpretable gesture prototypes and prototype abstractions make K-OMM-trees a well-suited method for interactive systems. Interestingly, the number of OMM model states as well as the variance hyper-parameter of OMMs have almost no influence on the classification performance of K-OMM-trees. Only variations in the tree's order leads to considerable decreases in accuracy. This conforms our analysis of the evolving data representation of K-OMM-trees which showed that the learned hierarchical structure reflects an inherent structure in the time series data. Thereby, higher levels in the tree are associated with more general gesture properties, whereas nodes closer to the leaves represent more specific gesture features. It this the extraction of this structure that enables better learning of instances of interactive behavior.

In our ongoing research we focus on applying the learned motion hierarchy to imitation learning in human-agent interaction scenarios. Additionally, we investigate the application of K-OMM-trees to further data domains.

Acknowledgments

This work has partially been supported by the Center of Excellence for Cognitive Interaction Technology (CITEC), funded by the German Research Foundation (DFG).

REFERENCES

[1] R. Amit and M. Mataric, 'Learning movement sequences from demonstration', in *ICDL '02: Proceedings of the 2nd Int. Conf. on Development and Learning*, pp. 203–208, Cambridge, Massachusetts, (2002). MIT Press.

[2] S. Calinon, F. D'halluin, E.L. Sauser, D.G. Caldwell, and A.G. Billard, 'Learning and reproduction of gestures by imitation', *Robotics Automation Magazine, IEEE*, **17**(2), 44 –54, (2010).

[3] S. Chiba and H. Sakoe, 'Dynamic programming algorithm optimization for spoken word recognition', *IEEE Transactions on Acoustics, Speech and Signal Processing*, **26**(1), 43, (1978).

[4] C. M. De Vries, L. De Vine, and S. Geva, 'Random Indexing K-tree', *ADCS 2009: Australian Document Computing Symposium 2009, Sydney, Australia*, 43–50, (2009).

[5] A. P. Dempster, N. M. Laird, and D. B. Rubin, 'Maximum likelihood from incomplete data via the em algorithm', *Journal of the Royal Statistical Society, Series B*, **39**(1), 1–38, (1977).

[6] J.D. Ferguson. Variable duration models for speech, 1980.

[7] S. Geva, 'K-tree: a height balanced tree structured vector quantizer', in *IEEE Neural Networks Signal Processing*, volume 1, pp. 271–280, (2000).

[8] U. Großekathöfer, A. Barchunova, R. Haschke, T. Hermann, M. Franzius, and H. Ritter, 'Learning of Object Manipulation Operations from Continuous Multimodal Input', in *Proceedings of the IEEE/RAS Int.Conf. on Humanoid Robots 2011*, (2011).

[9] U. Großekathöfer, A. Sadeghipour, T. Lingner, P. Meinicke, T. Hermann, and S. Kopp, 'Low Latency Recognition and Reproduction of Natural Gesture Trajectories', in *Proceedings of the 1st Int.Conf. on Pattern Recognition Applications and Methods (ICPRAM2012)*, pp. 154–161, (2012).

[10] U. Großekathöfer, N.-C. Wöhler, T. Hermann, and S. Kopp, 'On-the-fly behavior coordination for interactive virtual agents – A model for learning, recognizing and reproducing hand-arm gestures online', in *Proceedings of the 11th Int.Conf. on Autonomous Agents and Multiagent Systems (AAMAS2012)*, (2012).

[11] T. Inamura, I. Toshima, and Y. Nakamura, 'Acquiring motion elements for bidirectional computation of motion recognition and generation', in *Experimental Robotics VIII*, eds., B. Siciliano and P. Dario, volume 5, pp. 372–381. Springer-Verlag, (2003).

[12] M.T. Johnson, 'Capacity and complexity of HMM duration modeling techniques', *IEEE Signal Processing Letters*, **12**(5), 407–410, (2005).

[13] D. Kulić, W. Takano, and Y. Nakamura, 'Incremental learning, clustering and hierarchy formation of whole body motion patterns using adaptive hidden markov chains', *The Int. Journal of Robotics Research*, **27**(7), 761, (2008).

[14] J. Kwon and F.C. Park, 'Natural movement generation using hidden markov models and principal components', *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, **38**(5), 1184–1194, (2008).

[15] S.E. Levinson, 'Continuously variable duration hidden Markov models for automatic speech recognition', *Computer Speech & Language*, **1**(1), 29–45, (1986).

[16] J. MacQueen et al., 'Some methods for classification and analysis of multivariate observations', in *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, volume 1, p. 14. California, USA, (1967).

[17] L. R. Rabiner, 'A tutorial on hidden markov models and selected applications in speech recognition', *Proceedings of the IEEE*, **77**(2), 257–286, (Feb. 1989).

[18] D. Tolani, A. Goswami, and N.I. Badler, 'Real-time inverse kinematics techniques for anthropomorphic limbs', *Graphical models*, **62**(5), 353–388, (2000).

[19] SV Vaseghi, 'State duration modelling in hidden Markov models', *Signal processing*, **41**(1), 31–41, (1995).

[20] A.D. Wilson and A.F. Bobick, 'Parametric hidden markov models for gesture recognition', *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, **21**(9), 884–900, (1999).

[21] N.-C. Wöhler, U. Großekathöfer, A. Dierker, M. Hanheide, S. Kopp, and T. Hermann, 'A calibration-free head gesture recognition system with online capability', in *Proceedings of the 10th Int. Conf. on Pattern Recognition (ICPR2010)*, pp. 3814–3817. IEEE Computer Society, (2010).

Author Index