Proceedings eNTERFACE'12

The 8th International Summer Workshop on Multimodal Interfaces

July 2nd - July 27th 2012 ; Supélec, Metz, France

Prof. Olivier Pietquin, Chair

Supélec
Metz Technopôle
2 rue Edouard Belin
57070 Metz, France

Ph.: +33 (0)3 87 76 47 70
Fax : +33 (0)3 87 76 47 00
Olivier.Pietquin@Supelec.fr
http://malis.metz.supelec.fr/~pietquin
http://enterface12.metz.supelec.fr

Forewords

The eNTERFACE'12 workshop was organized by the Metz' Campus of Supélec and co-sponsored by the ILHAIRE and Allegro European projects.

The previous workshops in Mons (Belgium), Dubrovnik (Croatia), Istanbul (Turkey), Paris (France), Genoa (Italy), Amsterdam (The Netherlands) and Plzen (Czech Republic) had an impressive success record and had proven the viability and usefulness of this original workshop. eNTERFACE'12 hosted by Supélec in Metz (France) took this line of fruitful collaboration one step further. Previous editions of eNTERFACE have already inspired competitive projects in the area of multimodal interfaces, has secured the contributions of leading professionals and has encouraged participation of a large number of graduate and undergraduate students.

We received high quality project proposals among which the 8 following projects were selected.

1. Speech, gaze and gesturing - multimodal conversational interaction with Nao robot
2. Laugh Machine
3. Human motion recognition based on videos
4. M2M -Socially Aware Many-to-Machine Communication
5. Is this guitar talking or what!?
6. CITYGATE, The multimodal cooperative intercity Window
7. Active Speech Modifications
8. ArmBand : Inverse Reinforcement Learning for a BCI driven robotic arm control

All the projects resulted in promising results and demonstrations which are reported in the rest of this document. The workshop gathered more than 70 attendees coming from 16 countries all around Europe and even further. We received 4 invited speakers (Laurent Bougrain, Thierry Dutoit, Kristiina Jokinen and Anton Batliner) whose talks were greatly appreciated. The workshop was held in a brand new 800 m2 building in which robotics materials as well as many sensors were available to the attendees. This is why we proposed a special focus of this edition on topics related to human-robot and human-environment interaction. This event was a unique opportunity for students and experts to meet and work together, and to foster the development of tomorrow's multimodal research community.

All this has been made possible thanks to the the good will of many of my colleagues who volunteered before and during the workshop. Especially, I want to address many thanks to Jérémy

who did a tremendous job for making this event as enjoyable and fruitful as possible. Thanks a lot to Matthieu, Thérèse, Danièle, Jean-Baptiste, Senthil, Lucie, Edouard, Bilal, Claudine, Patrick, Michel, Dorothée, Serge, Calogero, Yves, Eric, Véronique, Christian, Nathalie and Elisabeth. Organizing this workshop was a real pleasure for all of us and we hope we could make it a memorable moment of work and fun.

Olivier Pietquin

Chairman of eNTERFACE'12

The eNTERFACE'12 Sponsors

We want to express our gratitude to all the organizations which made this event possible.

The eNTERFACE'12 Scientific Committee

Niels Ole Bernsen, University of Southern Denmark - Odense, Denmark
Thierry Dutoit, Faculté Polytechnique de Mons, Belgium
Christine Guillemot, IRISA, Rennes, France
Richard Kitney, University College of London, United Kingdom
Benoît Macq, Université Catholique de Louvain, Louvain-la-Neuve, Belgium
Cornelius Malerczyk, Zentrum für Graphische Datenverarbeitung e.V, Germany
Ferran Marques, Univertat Politécnica de Catalunya PC, Spain
Laurence Nigay, Université Joseph Fourier, Grenoble, France
Olivier Pietquin, Supélec, Metz, France
Dimitrios Tzovaras, Informatics and Telematics Intsitute, Greece
Jean-Philippe Thiran, Swiss Federal Institute of Technology Lausanne (EPFL), Switzerland
Jean Vanderdonckt, Université Catholique de Louvain, Louvain-la-Neuve, Belgium

The eNTERFACE'12 Local Organization Committee

General chair	Olivier Pietquin
Co-chair	Jeremy Fix
Web management	Claudine Mercier
Technical support	Jean-Baptiste Tavernier
Social activities	Matthieu Geist
Administration	Danielle Cebe
	Thérèse Pirrone

eNTERFACE 2012 - Project reports

Speech, gaze and gesturing: multimodal conversational interaction with Nao robot

Adam Csapo, Emer Gilmartin, Jonathan Grizou, JingGuang Han, Raveesh Meena, Dimitra Anastasiou, Kristiina Jokinen, and Graham Wilcock

***Abstract*—The report presents a multimodal conversational interaction system for the Nao humanoid robot, developed by project P1 at eNTERFACE 2012. We implemented WikiTalk, an existing spoken dialogue system for open-domain conversations, on Nao. This greatly extended the robot's interaction capabilities by enabling Nao to talk about an unlimited range of topics. In addition to speech interaction, we developed a wide range of multimodal interactive behaviours by the robot, including face-tracking, nodding, communicative gesturing, proximity detection and tactile interrupts. We made video recordings of user interactions and used questionnaires to evaluate the system. We further extended the robot's capabilities by linking Nao with Kinect.**

***Index Terms*—human-robot interaction, spoken dialogue systems, communicative gesturing.**

I. Introduction

The report presents a multimodal conversational interaction system for the Aldebaran Nao humanoid robot, developed by project P1 at eNTERFACE 2012. Our project's starting point was a speech-based open-domain knowledge access system. By implementing this system on the robot, we greatly extended Nao's interaction capabilities by enabling the robot to talk about an unlimited range of topics. In addition to speech interaction, we developed a wide range of multimodal interactive behaviours by the robot, including face-tracking, nodding, communicative gesturing, proximity detection and tactile interrupts, to enhance naturalness, expressivity, user-friendliness, and add liveliness to the interaction.

As the basis for speech interaction, we implemented on Nao the WikiTalk system [1], [2], that supports open-domain conversations using Wikipedia as a knowledge source. Earlier work with WikiTalk had used a robotics simulator. This report describes the multimodal interactive behaviours made possible by implementing "Nao WikiTalk" on a real robot.

Based on the above, the Nao robot with Nao WikiTalk can be regarded as a cognitive robot, since it can reason about how to behave in response to the user's actions. However, in the broader sense, the combination of Nao and WikiTalk is also viewed as a cognitive infocommunication system, as it allows users to interact via the robot with Wikipedia content that is remote and maintained by a wider community.

This report was published at CogInfoCom 2012 [3].

A. Csapo is with Budapest University of Technology and Economics.
E. Gilmartin and J. Han are with Trinity College Dublin.
J. Grizou is with INRIA, Bordeaux.
R. Meena is with KTH, Stockholm.
D. Anastasiou is with University of Bremen.
K. Jokinen and G. Wilcock are with University of Helsinki. e-mail: kristiina.jokinen@helsinki.fi, graham.wilcock@helsinki.fi.

The report is structured as follows. Section II explains the multimodal capabilities that we developed for Nao, focussing on communicative gesturing and its integration with speech interaction. Section III describes the system architecture and Section IV presents an evaluation of the system based on questionnaires and video recordings of human-robot interactions. Finally, Section V introduces the use of Kinect with Nao to further extend interaction functionality.

II. Multimodal Capabilities

Human face-to-face interaction is multimodal, involving several input and output streams used concurrently to transmit and receive information of various types [4]. While propositional content is transmitted verbally, much additional information can be communicated via non-verbal and paralinguistic audio ('um's and 'ah's in filled pauses, prosodic features), and visual channels (eye-gaze, gesture, posture). These non-verbal signals and cues play a major part in management of turn-taking, communicating speaker and listener affect, and signaling understanding or breakdown in communication.

During interaction speakers and listeners produce bodily movements which, alone or in tandem with other audio and visual information, constitute cues or signals which aid understanding of linguistic information, signal comprehension, or display participants' affective state. Movements include shifts in posture, head movements, and hand or arm movements. We take 'gesture' to include head and hand or arm movements.

A. Gestures

Nao Wikitalk was designed to incorporate head, arm and body movements to approximate gestures used in human conversation. This section describes the motivation for adding gestures to Nao, and their design and synthesis. A more comprehensive description of enhancing Nao with gestures and posture shifts can be found in [5].

Gestures take several forms and perform different functions. Following [6], we can distinguish commands and communicative gestures, and the latter can be categorized further as speech-independent (emblems -'ok' sign) or speech dependent (gestures accompanying speech). Speech dependent gestures may be iconic or metaphoric - "the fish was this big" with hands apart to show dimension, a palm-upward 'giving' gesture at start of narration. They may also be deictic (pointing to real or virtual objects) or beat gestures (simple flicks which mark time on speech) [7]. Nods and eye gaze movements are also visual cues to turn-taking management and

Gesture	Purpose	Description
Open hand palm up	Presentation of new paragraph	The gestures mimics the offering of information to the subject.
Open hand palm vertical	Presentation of new information	Up and down movement to mark new piece of information.
Head nod down	Indicating end of sentence	Upon seeing links in a sentence. To mark new info.
Head nod up	Indicating surprise	On being interrupted.
Speaking to standing	Listening mode	Nao goes to standing pose and listens to speaker.
Standing to speaking	Speaking mode	Nao goes to speaking pose when speaking.

TABLE I
NON-VERBAL GESTURES AND THEIR ROLE IN INTERACTION WITH NAO

comprehension in speakers and listeners with listeners nodding feedback, and speakers using upsweeps and gazing at listeners to check understanding and invite contributions/feedback [8].

Nao Wikitalk allows the user to query Wikipedia via the Nao robot and have chosen entries read out by the robot. In a text-free environment the user needs to infer the structure of the article from the robot's output - Wikipedia entries are large blocks of text which can be very monotonous when simply read out by a synthetic voice, and comprehension could be enhanced by adding non-verbal cues to discourse level organization of the text. In Wikipedia relevant information is marked with hyperlinks to other entries. A system where the robot could signal these links non-verbally while reading the text would allow the user to further query the encyclopedia without recourse to explicit menus. Gesture and posture changes could also be used to help manage turntaking in Nao's dialogue, while the inclusion of gesture in Nao's conversational repertoire would also enhance expressivity and add liveliness to the interaction.

As a first step towards adding these functionalities to Nao, we identified a set of gestures which could be used to:

- Mark discourse level details such as paragraph and sentence boundaries.
- Indicate hyperlinks
- Help manage turntaking
- Add expressivity or liveliness

Table I provides an overview of the chosen gesture set.

B. Gesture synthesis

Gestures are performed as a sequence of actions, the most prominent of which is the key pose, which captures the essence of the gesture and conveys much of its communicative payload. The approach taken to gesture synthesis in Nao was to create an animation sequence which could start at any body pose, move to the key pose or action core, and then continue to a follow-up pose which would complete the gesture.

The gesture synthesis process began with the isolation of key poses in the gestures. These key poses were then created in the Nao manually and their parameters recorded using Nao's Choregraphe animation software. The key poses that we have defined for the purpose of this work are shown in Figures A to G in Figure 1. To illustrate, Figure C specifies the key pose for the open hand palm up gesture.

The gestures were then created using Choregraphe's stop motion animation tools to interpolate the position of the robot's joints between the poses comprising the gesture. For example, the open hand palm up gesture for paragraph beginning was synthesized as an interpolated animation of the following sequence of key poses: *Standing→Speaking→Open-hand Palm-up→Speaking*. In a similar fashion an emphatic beat gesture was synthesized as an interpolated animation of the sequence: *Speaking→Open-hand Palm-vertical→Speaking*. The sequence *Open-hand Palm-vertical→Speaking* could be animated in a loop for synthesizing rhythmic beat gestures for a sequence of new information. The gestures thus created could then be programmed into the robot for later performance.

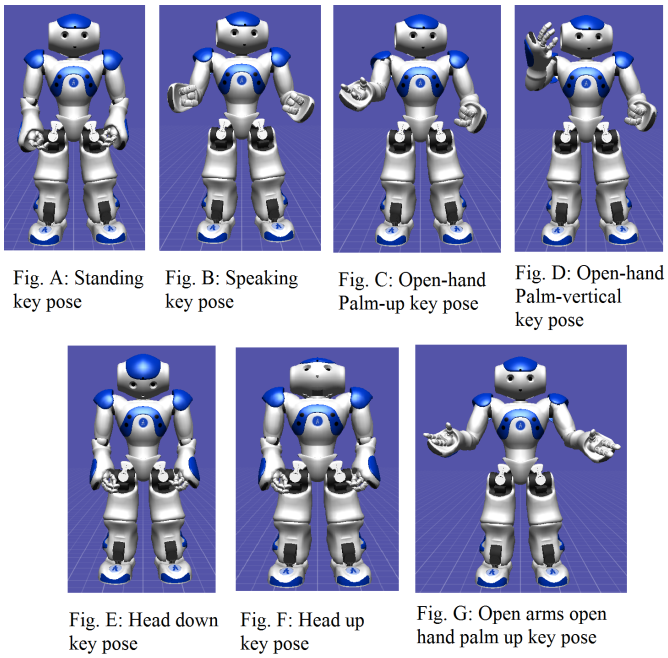


Fig. 1. Key poses.

During the animation process it became evident that the animation software did not accurately reflect the timing of gestures when performed by the robot rather than onscreen. This reflects the mechanical limitations of the motors of the robot. In order to better control the timing of gestures and to add flexibility to the robot dynamics we obtained the corresponding Python code for each gesture and defined the gestures as parameterized functions. In this way gesture duration and speed could be finely controlled from the Wikitalk code rather than called as monolithic action sequences.

C. Synchronizing gestures with speech

The gesture sequences created for the Nao accompany speech. To create an illusion of coherence requires fine timing

control and synchronization of the gesture with the relevant utterance - ideally aligning the gesture peak with the pitch accent of the marked word or phrase. A model for this sophisticated synthesis could not be explored given the rather short duration of the workshop. Instead we took the approach of synthesizing gestures with rather generic parameters so that they would not be perceived completely out of place.

In the system, gesture is controlled by a Gesture Manager (GM). The GM first identifies the relevant gesture for the planned utterance, using contextual details such as the status of discourse, the dialogue context and the contextual information in the article. The GM marks up the utterance to be spoken with tags containing information about the type of gesture that is to be triggered. The utterance and accompanying gesture are then created by the speech and the gesture synthesis components and sent to be executed by the robot.

The system currently includes gestures to mark discourse and structural features in the spoken text, and to signal the presence of new information at hyperlinks, both adding liveliness to the dialogue. We had intended to explore the turn taking mechanism in dialogue using gestures and gaze, but the Nao speech recognizer did not allow barge-in, in effect forcing the user to wait for a 'beep' before responding. Therefore, although the presence of a natural upsweep of the head at turn ceding by the Nao worked very well in prompting the user to speak, it was counterproductive in the Nao's current implementation as the user would speak 'before the beep' and thus before ASR had been enabled, confusing rather than enhancing the interaction. It was also noted that the motors were not always fast enough to produce gestures at the precise time indicated. Both of these problems are the result of engineering limitations, and it is highly likely that newer robots will offer improved performance, allowing a fuller range of gesture to be implemented in the system, and improving the timing of currently implemented gestures.

D. Face detection, tactile sensors, and non-verbal cues

As non-verbal information is vital in human face to face interaction, it is desirable for an anthropomorphic embodied conversational agent (ECA) to have facilities to synthesise and recognize non-verbal audio and visual information in addition to its speech synthesis and recognition modules. In this section we summarise the different methods and technologies that we studied for the Nao WikiTalk. The studies and experiments are discussed in more detail in [9].

The Nao platform provides several built-in technologies to enable non-verbal human-robot interaction. Using the Viola-Jones algorithm [10], Nao can detect faces and track people as well as detect the user's head movements like nodding and shakes. However, these capabilities interfere with other modules that send commands to the same motor, e.g. requests to nod, and the head movement appears "jerky" due to conflicting signals. We overcame this problem by deploying conflicting modules into separate threads.

We explored the use of sonar sensors and speech direction detection as conversation triggers. The robot can infer if there are users close by who may want to start a conversation.

Using sonar sensors, we recorded the distance between humans and robots in interactive situations, and could thus empirically test what is the optimal distance for human-robot interactions. In our setup, the best communication distance is about 0.9 meters.

Finally, we investigated different methods for interrupting the conversation, using tactile sensors and an object recognition method. The sensor on Nao's head was adopted as the most reliable method: when the user wants to interrupt Nao's speaking, he or she simply touches the robot on his head.

III. System Architecture

An overview of the system architecture is shown in Figure 2. At the heart of the system is a conversation manager, which consists of a finite state machine, and a number of interactive extensions that store various parameters of the user's past interactions and influence the functionality of the state machine accordingly. The conversation manager communicates with a Wikipedia manager on the one hand (so as to be able to obtain appropriately filtered text from Wikipedia), and a Nao manager on the other (so as to be able to map its states onto the actions of the Nao robot).

In order to enable the Nao robot to react to various events while reading text from Wikipedia, the Nao manager is capable of registering events and alerting the appropriate components of the system when anything of interest (either on the inside or the outside of the system) occurs. Figure 2 shows three examples of event handling within the Nao Talk module (the class which implements this module is directly connected to the Nao robot and drives its speech functionality). Functions isSaying(), startOfParagraph(), and endOfSentence() are all called periodically by the Nao manager, and return True whenever the robot stops talking, reaches the start of a paragraph, or finishes a sentence, respectively. Whenever such events occur, the Nao manager can trigger appropriate reactions, for example, through the Gestures module.

A. Interactive extensions within the conversation manager

The history of the user's interactions is stored in a statistics structure within the conversation manager. Using a set of simple heuristics, it is possible to create more interesting dialogues between the user and the robot by:

- ensuring that the robot does not give the same instructions to the user in the same way over and over again
- varying the level of sophistication in the functionalities that are introduced to the user by the robot. For example, in the beginning the robot gives simple instructions, allowing the user to practice and understand the basic functionalities of the system; for more advanced users, the system suggests new kinds of use cases which may not have previously been known to the user.

B. Events and event listeners in the Nao manager

As mentioned earlier, the Nao manager component is capable of registering and listening to events that occur either on the outside of the system, or within the system. Internal events related to speech synthesis include:

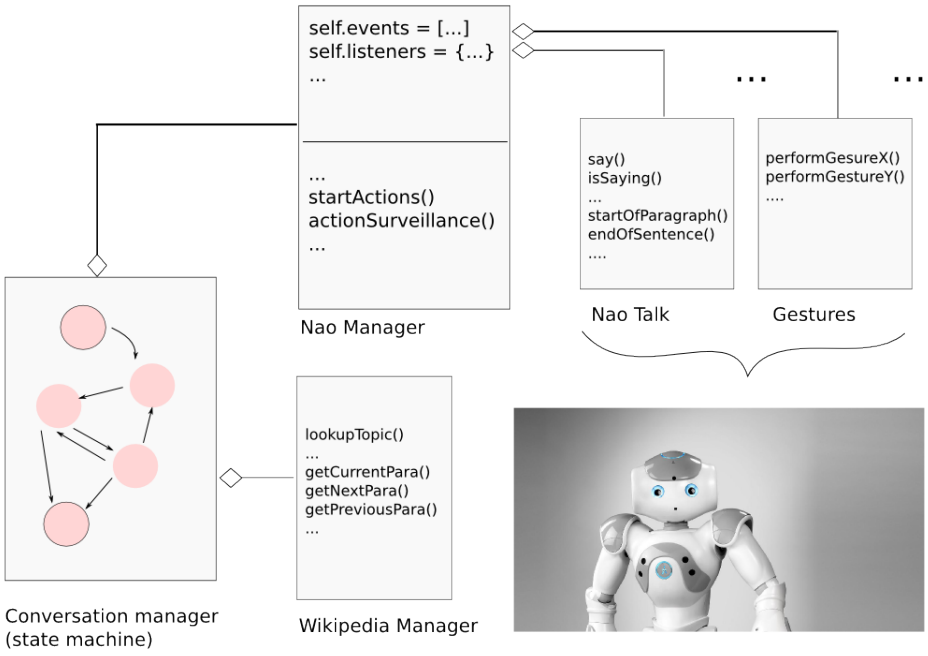


Fig. 2. Overall view of the system architecture.

- The start of new paragraph within the text
- The end of a sentence within the text
- The end of a logically coherent part of the text (for example, the end of a paragraph or a topic)
- The existence of a link within the text

External events related to the user's actions include:

- The user's proximity to the Nao robot's sonar sensors
- The user touching any of the 3 tactile sensors on the head of the Nao robot

The Nao manager can also be said to include implicit event listeners, which are an integral part of the Nao robot and need not be implemented explicitly by the developer. Examples of event listeners of this type include the Nao robot's capability to detect the presence of the user, track the user's head movements, or recognize the direction of a sound (e.g., when the user claps or makes other noises).

IV. User evaluation

To evaluate the impact of the various gestures and body movements exhibited by Nao during an interaction, we conducted a user evaluation of the system. Subjects were asked to take part in three 5-minute interactions. The subjects were told that Nao can provide them information from Wikipedia.

We followed the evaluation scheme proposed in [11]. Users were first asked to fill a questionnaire, which was designed to gauge their expectations from the system. After the interaction with the system the users filled in another questionnaire that gauged their experience with the system. We evaluated the system along the following dimensions: Interface, Responsiveness, Expressiveness, Usability and Overall experience. Before their first interaction with the system each user filled in a questionnaire about their expectations from the system. By doing so we subtly primed the user's attention to aspects of the conversation we wanted to evaluate. After each of the three interactions the users filled in another questionnaire regarding their experience. For each question participants were asked to provide their response on a five point scale (where 1: Strongly disagree and 5: Strongly agree). Table II illustrates the questionnaire for evaluating the user expectations and experience on robot gestures and body movements.

Twelve users participated in the evaluation. All of them were participants of the 8th International Summer Workshop on Multimodal Interfaces, eNTERFACE-2012. The subjects were given instructions to talk to Nao as much as they wish, and try out how well it can present them with interesting information. There were no constraints or restrictions on the topics. Users could ask Nao to talk about almost anything. In addition to this they were provided a list of commands to help them familiarize with the interaction control.

Figure 3 provides an overview of user expectations and their experiences on the questions presented in Table II. The user evaluation is discussed in more detail in [12].

V. Extending Nao with Kinect

Using Nao's own speech, sensing and acting capabilities makes the system easy to configure However we reached some of the limits of the Nao abilities, especially when it comes to detecting user behaviours Gesture recognition, gaze tracking or multiple interlocutors detection are currently beyond the embedded hardware and software of the Nao.

In order to enable more advanced interaction, we started to develop Kinect-based tools that can gather more precise data about the user's behaviour at the cost of an additional

System Aspect	Ref.	Expectation	Experience
Interface	I2	I expect to notice if Nao's hand gestures are linked to exploring topics.	I noticed Nao's hand gestures were linked to exploring topic.
Interface	I3	I expect to find Nao's hand and body movement distracting.	Nao's hand and body movement distracted me.
Interface	I4	I expect to find Nao's hand and body movements creating curiosity in me.	Nao's hand and body movements created curiosity in me.
Expressiveness	E1	I expect Nao's behaviour to be expressive	Nao's behaviour was expressive
Expressiveness	E2	I expect Nao will appear lively.	Nao appeared lively.
Expressiveness	E3	I expect Nao to nod at suitable times	Nao nodded at suitable times
Expressiveness	E5	I expect Nao's gesturing will be natural.	Nao's gesturing was natural.
Expressiveness	E6	I expect Nao's conversations will be engaging	Nao's conversations was engaging
Responsiveness	R6	I expect Nao's presentation will be easy to follow.	Nao's presentation was easy to follow.
Responsiveness	R7	I expect it will be clear that Nao's gesturing and information presentation are linked.	It was clear that Nao's gesturing and information presentation were linked.
Usability	U1	I expect it will be easy to remember the possible topics without visual feedback.	It was easy to remember the possible topics without visual feedback.
Overall	O2	I expect I will like Nao's gesturing.	I liked Nao's gesturing.
Overall	O3	I expect I will like Nao's head movements.	I liked Nao's head movements.

TABLE II
QUESTIONNAIRE FOR EVALUATING USER EXPECTATIONS AND EXPERIENCE WITH NAO.

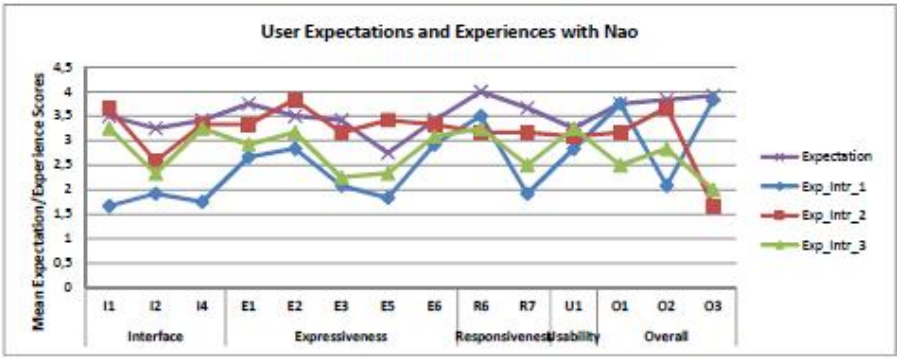


Fig. 3. User expectations and experiences with Nao.

external device. Microsoft Kinect is an inexpensive non-invasive technology which by means of a standard camera and a depth sensor is able to determinate the location of particular body joints in a 3D space. This section explains how it could be used to enhance the interaction with the Nao robot.

A. Application

Among the different potential applications of Kinect in our system, we distinguish three categories : (1) information that helps the robot understand the behaviour of the user and enhance the interaction, (2) information that helps us evaluate human-robot interaction during user experiments and (3) tools that help us enhance the behaviour of the robot.

1) Enhancing interaction: The face tracking option provide head orientation and position from which can be extracted an approximation of the gaze of the user. This information can be useful to detect if the user is bored during the interaction and trigger adapted robot behaviours, such as ending the topic, asking for a new topic... The skeleton tracking can be used to detect if a person enters or leaves the room as well as their position in the room. That could trigger welcome and goodbye behaviour as well as focus the gaze of the robot in the direction of the user. (Note that the face tracking ability already included with Nao robots is limited to close range and proper light interaction, the Kinect is more robust to ambient condition and allows for a larger interaction area.) A gesture recognition module using data from the Kinect [13] would enable non-verbal communication between human and robot. In our current set-up, the robot quite often uses confirmation questions that can be boring for a user to verbally reply in the long run. The kind of recognizable gestures we could think of are nodding to say 'Yes' or 'No', arm movement to continue or stop the current topic. We could also use gesture data to focus the robot gaze towards the hands of the user when they perform a gesture. Kinect's multiple skeleton and face tracking abilities can even extend this to a multi-users setting.

2) Tracking user behaviours: Similar data can be used to track the user behaviour during an interaction in order to get quantitative measurements of the gaze of the user, the user restlessness, the talking position and so on.

3) Enhancing the behaviour of the robot: Using the Kinect, one could also think of tele-operating the Nao robot, meaning that the gesture of a human standing in front of a Kinect is mapped to the body of the robot. This would decrease the amount of work needed to develop gestures for the robot. Instead of blind trial and error sessions using a graphical representation of the joint evolution in time, one could directly record a gesture by 'demonstrating' it to the robot. [14] investigates the creation of an affect space for emotional body

language to be displayed by robots. The body postures were generated by means of motion capture data. This work focuses on static posture but can be extended to dynamic gesturing.

Finally, tele-operating the robot would make easier Wizard-of-Oz experiments where the robot gestures are remotely operated by an expert while a user experiment is running.

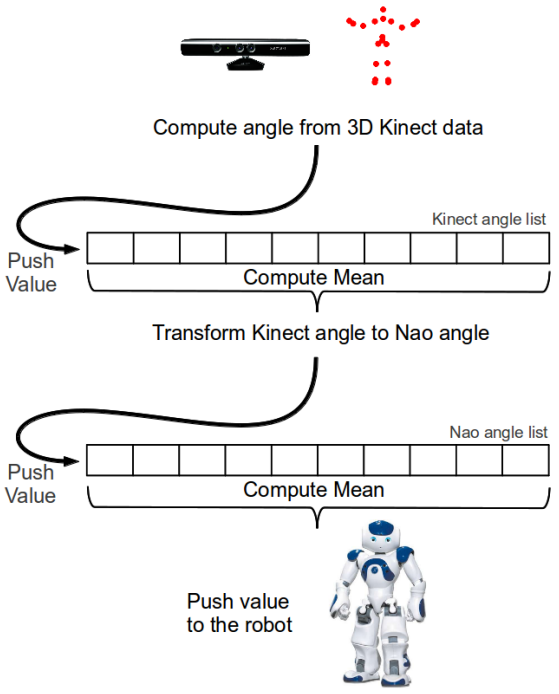


Fig. 4. Double mean filtering of the Kinect data.

B. Teleoperating Nao upper body using Kinect

In order to teleoperate the robot we need to extract useful angle values from the joint positions as well as to filter out the noise in the data received by the Kinect.

1) Extracting useful data: In order to map data from the Kinect to the Nao, we need to extract the corresponding angles from the skeleton points gathered though the Kinect. Two aspects have to be considered, (1) the angle measure have to be independent to any other movement of the human and (2) angles should correspond to one degree of freedom of the robot. As gathered data are points in a three-dimensional space, we have to choose the plane where points will be projected for the angle measurement.

2) Mapping: Depending on the reference and positive and negative direction, angles extracted from the Kinect data have to be shifted and/or inverted as well as min/max constrained to match with the particular Nao angle reference. This mapping depends on the points chosen and the positive direction defined. In our case we use a simple linear mapping from Kinect angle to Nao angle. A non linear mapping could also be used to have more precise movement in certain range.

3) Filtering: Data from Kinect are noisy. In order to get a smooth mapping from human gestures to robot movements, the noise has to be cancelled. Removing noise will add a delay between data acquisition and actual movement on the robot.

As shown in Figure 4, we use two mean filters in a row. For every new data from the Kinect, angles are computed and pushed into a list. The mean from this list is used to compute the corresponding Nao angle which is pushed into a second list. The mean of this Nao angle list is used to control the robot. The best buffer size was chosen by empirical tests.

If empty or incomplete data are received from the Kinect (person left the room, Kinect obstruction), an empty value is pushed into the Kinect angle list. This simple method allows a smooth and yet reactive filtering. In addition, we set fraction_of_max_speed to 0.5. This avoids the robot reaching its current goal before receiving a new one (i.e. avoid shaky movements) and has been evaluated by empirical tests.

ACKNOWLEDGMENT

The authors thank the organizers of eNTERFACE 2012 at Supelec, Metz for the excellent environment for this project.

REFERENCES

[1] K. Jokinen and G. Wilcock, "Constructive interaction for talking about interesting topics," in *Proceedings of Eighth International Conference on Language Resources and Evaluation (LREC 2012)*, Istanbul, 2012.

[2] G. Wilcock, "WikiTalk: A spoken Wikipedia-based open-domain knowledge access system," in *Proceedings of the COLING 2012 Workshop on Question Answering for Complex Domains*, Mumbai, India, 2012, pp. 57–69.

[3] A. Csapo, E. Gilmartin, J. Grizou, J. Han, R. Meena, D. Anastasiou, K. Jokinen, and G. Wilcock, "Multimodal conversational interaction with a humanoid robot," in *Proceedings of 3rd IEEE International Conference on Cognitive Infocommunications (CogInfoCom 2012)*, Kosice, 2012.

[4] J. Allwood, "Bodily Communication – Dimensions of Expression and Content," in *Multimodality in Language and Speech Systems*, B. Granström, D. House, and I. Karlsson, Eds. Kluwer Academic Publishers, Dordrecht, 2002, pp. 7–26.

[5] R. Meena, K. Jokinen, and G. Wilcock, "Integration of gestures and speech in human-robot interaction," in *Proceedings of 3rd IEEE International Conference on Cognitive Infocommunications (CogInfoCom 2012)*, Kosice, 2012.

[6] F. Quek, "Toward a vision-based hand gesture interface," in *Proceedings of the Virtual Reality System Technology Conference*, Singapore, 1994, pp. 17–29.

[7] A. Kendon, *Gesture: Visible action as utterance*. Cambridge University Press, 2004.

[8] C. Navarretta, E. Ahlsén, J. Allwood, K. Jokinen, and P. Paggio, "Feedback in Nordic first-encounters: a comparative study," in *Proceedings of Eighth International Conference on Language Resources and Evaluation (LREC 2010)*, Istanbul, 2012.

[9] J. Han, N. Campbell, K. Jokinen, and G. Wilcock, "Investigating the use of non-verbal cues in human-robot interaction with a Nao robot," in *Proceedings of 3rd IEEE International Conference on Cognitive Infocommunications (CogInfoCom 2012)*, Kosice, 2012.

[10] P. Viola and M. J. Jones, "Robust real-time face detection," *International journal of computer vision*, vol. 57, no. 2, pp. 137–154, 2004.

[11] K. Jokinen and T. Hurtig, "User expectations and real experience on a multimodal interactive system," in *Proceedings of Ninth International Conference on Spoken Language Processing (Interspeech 2006)*, Pittsburgh, USA, 2006.

[12] D. Anastasiou, K. Jokinen, and G. Wilcock, "Evaluation of WikiTalk - user studies of human-robot interaction," in *Proceedings of 15th International Conference on Human-Computer Interaction (HCII 2013)*, Las Vegas, USA, 2013.

[13] K. Lai, J. Konrad, and P. Ishwar, "A gesture-driven computer interface using Kinect," in *Image Analysis and Interpretation (SSIAI 2012)*, 2012, pp. 185–188.

[14] A. Beck, L. Canamero, and K. Bard, "Towards an affect space for robots to display emotional body language." in *RO-MAN, 2010 IEEE*, 2010, pp. 464–469.

Laugh Machine

Jérôme Urbain[1], Radoslaw Niewiadomski[2], Jennifer Hofmann[3], Emeline Bantegnie[4], Tobias Baur[5], Nadia Berthouze[6], Hüseyin Çakmak[1], Richard Thomas Cruz[7], Stéphane Dupont[1], Matthieu Geist[10], Harry Griffin[6], Florian Lingenfelser[5] Maurizio Mancini[8], Miguel Miranda[7], Gary McKeown[9], Sathish Pammi[2], Olivier Pietquin[10], Bilal Piot[10], Tracey Platt[3], Willibald Ruch[3], Abhishek Sharma[2], Gualtiero Volpe[8] and Johannes Wagner[5]

[1]TCTS Lab, Faculté Polytechnique, Université de Mons, Place du Parc 20, 7000 Mons, Belgium
[2]CNRS - LTCI UMR 5141 - Telecom ParisTech, Rue Dareau, 37-39, 75014 Paris, France
[3]Universität Zürich, Binzmuhlestrasse, 14/7, 8050 Zurich, Switzerland
[4]LA CANTOCHE PRODUCTION, Hauteville, 68, 75010 Paris, France
[5]Institut für Informatik, Universität Augsburg, Universitätsstr. 6a, 86159 Augsburg, Germany
[6]UCL Interaction Centre, University College London, Gower Street, London, WC1E 6BT, United Kingdom
[7]Center for Empathic Human-Computer Interactions, De la Salle University, Manila, Philippines
[8]Universita Degli Studi di Genova, Viale Francesco Causa, 13, 16145 Genova, Italy
[9]The Queen's University of Belfast, University Road, Lanyon Building, BT7 1NN Belfast, United Kingdom
[10]École Supérieure d'Électricité, Rue Edouard Belin, 2, 57340 Metz, France

***Abstract*—The Laugh Machine project aims at endowing virtual agents with the capability to laugh naturally, at the right moment and with the correct intensity, when interacting with human participants. In this report we present the technical development and evaluation of such an agent in one specific scenario: watching TV along with a participant. The agent must be able to react to both, the video and the participant's behaviour. A full processing chain has been implemented, integrating components to sense the human behaviours, decide when and how to laugh and, finally, synthesize audiovisual laughter animations. The system was evaluated in its capability to enhance the affective experience of naive participants, with the help of pre and post-experiment questionnaires. Three interaction conditions have been compared: laughter-enabled or not, reacting to the participant's behaviour or not. Preliminary results (the number of experiments is currently to small to obtain statistically significant differences) show that the interactive, laughter-enabled agent is positively perceived and is increasing the emotional dimension of the experiment.**

***Index Terms*—Laughter, virtual agent.**

I. Introduction

LAUGHTER is a significant feature of human communication, and machines acting in roles like companions or tutors should not be blind to it. So far, limited progress has been made towards allowing computer-based applications to deal with laughter. In consequence, only few interactive multimodal systems exist that use laughter in the interactions. Within the long term aim of building a truly interactive machine able to laugh and respond to human laughter, during the eNTERFACE Summer Workshop 2012 we have developed the Laugh Machine project.

This project had three main objectives. First of all we aimed to build an interactive system that is able to detect the human laughs and to laugh back appropriately (*i.e.*, right timing, right type of laughter) to the human and the context. Secondly, we wanted to use the laughing agent to support psychological studies investigating benefits of laughter in human-machine interaction and consequently improve the system towards more naturalness and believeability. The third aim was the collection of multimodal data on human interactions with the agent-based system.

To achieve these aims, we tuned and integrated several existing analysis components that can detect laughter events as well as interpreters that controlled how the virtual agent should react to them. In addition, we also provided output components that are able to synthesize audio-visual laughs. All these components were integrated to work in real-time. Secondly, we focused on building an interactive scenario where our laughing agent can be used. In our scenario, the participant watches a funny stimulus (*i.e.*, film clip, cartoon) together with the virtual agent. The agent is able to laugh, reacting to both, the stimulus and the user's behavior. We evaluated the impact of the agent through user evaluation questionnaires (*e.g.*, assessing the mood pre and post experiments, funniness and aversiveness ratings to both stimuli and agent behavior, etc.). At the same time we were able to collect multimodal data (audio, facial expressions, shoulder movements, and Kinect depth maps) of people interacting with the system.

This report is organized as follows. First, related work is presented in Section II. Then, the experimental scenarios are outlined in Section III, so that the framework for developing the technical setup is known. The data used for training the components is presented in section IV. Section V shows the global architecture of the Laugh Machine system. The next sections focus on the components of this system: details about the input components are given in Section VI, Section VII is related to the dialog manager and the output components are described in Section VIII. Then, the conducted experiments to evaluate the system are explained in Section IX. The results of

these experiments are discussed in Section X. Section XI refers to the data that has been collected during the experiments. Finally, Section XII presents the conclusions of the project.

II. Related work

Building an interactive laughing agent requires tools from several fields: at least audiovisual laughter synthesis for the output, and components able to detect particular events like participant's laughs and decide when and how to laugh. In the following paragraphs we will present the main works in audiovisual laughter recognition, acoustic laughter synthesis and visual laughter synthesis, then the interactive systems involving laughter that have already been built. Regarding a decision component dealing with laughter as input and output, to the best of our knowledge there is no existing work.

A. Audiovisual laughter recognition

In the last decade, several systems have been built to distinguish laughter from other sounds like speech. It started with audio-only detection. The global approach followed up to now for discriminating speech and laughter is to compute standard acoustic features (MFCCs, pitch, energy, ...) and feed them into typical classifiers: Gaussian Mixture Models (GMMs), Support Vector Machines (SVMs) or Multi-Layer Perceptrons (MLPs). Kennedy and Ellis [1] obtained 87% of classification accuracy with SVMs fed with 6 MFCCs; Truong and van Leeuwen [2] reached slightly better results (equal error rate of 11%) with MLPs fed with Perceptual Linear Prediction features; Knox and Mirghafori [3] obtained better performance (around 5% of error) by using temporal feature windows (feeding the MLPs with the features belonging to the past, current and future frames).

In 2008, Petridis and Pantic started to enrich the so far mainly audio-based work in laughter detection by consulting audio-visual cues for decision level fusion approaches [4]–[6]. They combined spectral and prosodic features from the audio modality with head movement and facial expressions from the video channel. Results suggest that integrated information from audio and video leads to improved classification reliability compared to a single modality - even with fairly simple fusion methods. They reported a classification accuracy of 74.7% to distinguish three classes, namely unvoiced laughter, voiced laughter and speech. In [7] they present a new classification approach for discriminating laughter from speech by modelling the relationship between acoustic and visual features with Neural Networks.

B. Acoustic laughter synthesis

Acoustic laughter synthesis is an almost unexplored domain. Only 2 attempts have been reported in literature. Sundaram and Narayanan [8] modeled the laughter intensity rhythmic envelope with the equations governing an oscillating mass-spring and synthesized laughter vowels by Linear Prediction. This approach to laughter synthesis was interesting, but the produced laughs were judged as non-natural by listeners. Lasarcyk and Trouvain [9] compared laughs synthesized by an articulatory system (a 3D modeling of the vocal tract) and diphone concatenation. The articulatory system gave better results, but they were still evaluated as significantly less natural than human laughs. In 2010, Cox conducted an online evaluation study to measure to what extent (copy-)synthesized laughs were perceived as generated by a human or a computer [10]. Laughs synthesized by the 2 aforementioned groups were included in the study, as well as a burst-concatenation copy-synthesized laughter proposed by UMONS, which obtained the best results with almost 60% of the 6000 participants thinking it could be a human laugh. Nevertheless, this number is far from the 80% achieved by a true human laugh.

C. Visual laughter synthesis

The audio-synchronous visual synthesis of laughter requires the development of innovative hybrid approaches that combine several existing animation techniques such as data-driven animation, procedural animation and machine learning based animation. Some preliminary audio-driven models of laughter have been proposed. In particular Di Lorenzo et al. [11] proposed an anatomic model of torso respiration during laughter, while Cosker and Edge [12] worked on facial animation during laughter. The first model does not work in real-time while the second is limited to only facial animation.

D. Laughing virtual agents

Urbain et al. [13] have proposed the AVLaughterCycle machine, a system able to detect and respond to human laughs in real time. With the aim of creating an engaging interaction loop between a human and the agent they built a system capable of recording the user's laugh and responding to it with a similar laugh. The virtual agent response is automatically chosen from an audio-visual laughter database by analyzing acoustic similarities with the input laughter. This database is composed of audio samples accompanied by the motion capture data of facial expressions. While the audio content is directly replayed, the corresponding motion capture data are retargeted to the virtual model.

Shahid et al. [14] proposed Adaptive Affective Mirror, a tool that is able to detect user's laughs and to present audio-visual affective feedback, which may elicit more positive emotions in the user. In more details, Adaptive Affective Mirror produces a distortion of the audio-visual input using real-time graphical filters such as bump distortion. These distortions are driven by the amount and type of user's laughter that has been detected. Fukushima et al. [15] built a system able to increase users' laughter reactions. It is composed of a set of toy robots that shake heads and play preregistered laughter sounds when the system detects the initial user laughter. The evaluation study showed that the system enhances the users' laughing activity (*i.e.*, generates the effect of contagion).

Finally, Becker-Asano et al. [16] studied the impact of auditory and behavioral signals of laughter in different social robots. They discovered that the social effect of laughter depends on the situational context including the type of task executed by the robot, verbal and nonverbal behaviors (other than laughing) that accompany the laughing act [17]. They also

claim that inter-cultural differences exist in the perception of naturalness of laughing humanoids [16].

III. Scenarios and Stimulus Film

In our evaluation scenario the virtual agent and its laughter behavior were investigated. The experimental setup involved a participant watching a funny video with a virtual agent visually present on a separate screen. The expressive behavior of the virtual agent was varied among three conditions, systematically altering the degree of expressed appreciation of the clip (amusement) in verbal and non-verbal behavior, as well as different degrees of interaction with the participant's behavior. The three conditions are:

- "fixed speech": the agent is verbally expressing amusement at pre-defined times of the video
- "fixed laughter": the agent is expressing amusement through laughs at pre-defined times of the video
- "interactive laughter": the agent is expressing amusement through laughter, in reaction to both the stimulus video and the participant's behavior

Furthermore, participant related variables were assessed with self-report instruments and allowed for the investigation of the influence of mood and personality on the perception and evaluation of the virtual agent. This allowed for the control of systematic biases on the evaluation of the virtual agent, which are independent of its believability (*e.g.*, individuals with a fear of being laughed at perceive all laughter negatively). The impact of the agent was assessed by investigating the influence of the session on participant's mood, as well as by self-report questionnaires assessing the perception of the virtual agent and the participant's cognitions, beliefs and emotions.

The stimulus film consisted of five candid camera pranks with a total length of 8 minutes. The clips were chosen by one expert rater who screened a large amount of video clips (approximately 4 hours) and chose five representative, culturally unbiased pranks sections of approximately 1 to 2 minutes length. All pranks were soundless and consisted of incongruity-resolution humor.

IV. Data used for training

Several pieces of data have been used in the project, two existing databases and two datasets specifically recorded to develop Laugh Machine. These databases are briefly presented in this section.

A. The SEMAINE database

The SEMAINE database [18] was collected for the SEMAINE-project by Queen's University Belfast with technical support of the HCI^2 group of Imperial College London. The corpus includes recordings from users while holding conversations with an operator who adopts in sequence four roles designed to evoke emotional reactions. One of the roles (Poppy) being happy and outgoing often invokes natural and spontaneous laughter by the user. The corpus is freely available for research purpose and offers high-audio quality, as well as, frontal and profile video recordings. The latter is important as it allows incorporation of visual features, which is part of the future work of Laugh Machine.

Within Laugh Machine, the SEMAINE database has been used to design a framework for laughter recognition (see Section VI-B) and select the most relevant audio features for this task.

Even though laughter is included as a class in the transcriptions of the SEMAINE database, provided laughter annotation tracks turned out to be too coarse to be used in the Laugh Machine training process. Hence, 19 sessions (each about 4-7 minutes long), which were found to include a sufficient number of laughs, were selected and manually corrected.

B. The AVLaughterCycle database

Secondly, we used the AudioVisualLaughterCycle (AVLC) corpus [19] that contains about 1000 spontaneous audio-visual laughter episodes with no overlapping speech. The episodes were recorded with the participation of 24 subjects. Each subject was recorded watching a 10-minutes comedy video. Thus it is expected that the corpus contains mainly amusement laughter. Each episode was captured with one motion capture system (either Optitrack or Zigntrack) and synchronized with the corresponding audiovisual sample. The material was manually segmented into episodes containing just one laugh. The number of laughter episodes for a subject ranges from 4 to 82. The annotations also include phonetic transcriptions of the laughter episodes [20].

Within Laugh Machine, the AVLaughterCycle database has been used to design the output components (audiovisual laughter synthesis, see Section VIII).

C. Belfast interacting dyads

The first corpus recorded especially for Laugh Machine contains human-human interactions when watching the stimulus film (see Secton III). Two dyads (one female-female, one male-male) were asked to watch the film. The two participants were placed in two rooms; they watched the same film simultaneously on two separate LCD displays. They could also see the other participant's reaction as a small window with the other person view was placed on the top of the displayed content. The data contains the closeup view of each participant's face, 90 degree views (all at 50FPS) of the half of the body as well as audio tracks obtained from close-talk and far-field microphones for each participant, sampled at 48kHz and stored in PCM 24bits. Laughs have been segmented from the recorded signals.

This interaction data has been used to train the dialog manager component (see Section VII).

D. Augsburg scenario recordings

In order to tune the laughter detection (initially developed on the SEMAINE database) to the sensors actually used in Laugh Machine, a dedicated dataset has been recorded.

Since laughter includes respiratory, vocal, and facial and skeletomuscular elements [21], we can expect to capture signs of laughter if we install sensory to capture the user's voice,

facial expressions, and movements of the upper body. To have a minimum of sensors we decided to work with only two devices: the Microsoft Kinect and the Chest Band developed at the University College London (see Section VI-D). The latest version of the Microsoft Kinect SDK[1] not only offers full 3D body tracking, but also a real-time 3D mesh of facial features—tracking the head position, location of eyebrows, shape of the mouth, etc.

TABLE I
RECORDED SIGNALS.

Recording device	*Captured signal*	*Description*
Microsoft Kinect	Video	RGB, 30fps, 640x480
	Face points	
	Facial action units	
	Head pose	
	Skeleton joints	
	Audio	16 kHz, 16 bit, mono
Respiration Sensor	Thoracic circumference	120Hz, 8 bit

The recorded signals are summarized in Table I. Recordings took place at the University of Augsburg, using the Social Signal Interpretation (SSI, see Section VI-A) tool. During the sessions 10 German and 10 Arabic students were recorded while watching the stimulus film. By including participants with different cultural background it is our hope to improve the robustness of the final system. The recordings were then manually annotated at three levels: 1) beginning and ending of laughter in the audio track, 2) any non-laughter event in the audio track, such as speech and other noises, and 3) beginning and ending of smiles in the video track.

V. SYSTEM ARCHITECTURE

The general system architecture is displayed in Figure 1. We can distinguish 3 types of components: input components, decision components and output components. They are respectively explained in Sections VI, VII and VIII.

The input components are responsible for multimodal data acquisition and real-time laughter-related analysis. They include laughter detection from audiovisual features, body movements analysis (with laughter likelihood), respiration signal acquisition (also with laughter likelihood) and input laughter intensity estimation.

The decision components receive the information from the input components (*i.e.*, laughter likelihoods and intensity from multimodal features) as well as contextual information (*i.e.*, the funniness of the stimulus, see Section IX-C2, in green on Figure 1) and determines how the virtual agent should react. There are actually two decision components: the dialog manager, which decides if and how the agent should laugh at each time frame (typically 200ms), is followed by a block called "Laughter Planner", which decides whether or not the instruction to laugh should be forwarded to the synthesis components. In some cases, for example when there is an ongoing animation, it is indeed preferable not to transmit new synthesis instructions.

[1]http://www.microsoft.com/en-us/kinectforwindows/

The output components are responsible for the audiovisual laughter synthesis that is displayed when the decision components instruct to do so. In the current state of these components, it is not possible to interrupt a laughter animation (*e.g.*, to decide abruptly to stop laughing or on the other hand to laugh more intensely before the current output laughter is finished). This is the reason why the "Laughter Planner" module has been added. The Laugh Machine project includes one component for audio synthesis and 2 different animation Realizers, Greta and LivingActor (see Section VIII).

All the components have to work in real-time. Thus, the organization of the communication between different components is crucial in such project. For this purpose we use the SEMAINE[2] architecture which was originally aimed to build a Sensitive Listener Agent (SAL). The SEMAINE API is a distributed multi-platform component integration framework for real-time interactive systems. The architecture of SEMAINE API uses a message-oriented middleware (MoM) in order to integrate several components – where actual processing of the system is defined. Such components communicate via a set of topics. Here, a topic is a virtual channel where each and every published message, addressed to that topic, is delivered to its subscribed consumers. The communication passes via the message-oriented middleware ActiveMQ™ [22], which supports multiple operating systems and programming languages. For component integration, the SEMAINE API encapsulates the communication layer in terms of components that receive and send messages, and a system manager that verifies the overall system state, provides a centralized clock independent of the individual system clocks.

To integrate Laugh Machine components we used the same exchange messages server (*i.e.*, ActiveMQ) and the SEMAINE API. Each Laugh Machine component can read and write to some specific ActiveMQ topics. For this purpose we defined a hierarchy of message topics and for each topic the appropriate message format. Simple data (such as input data or clock signals) were coded in simple text messages in string/value tuples, so called *MapMessage*s, *e.g.* the message $AUDIO\_LAUGHTER\_DETECTION$ 1 is sent wherever laughter was detected from the audio channel. On the other hand more complex information such as the description of the behavior to be displayed was coded in standard XML languages such as the Behavior Markup Language[3] (BML).

It should be noted that, since the available data to train the decision components ((*i.e.*, the Belfast dyads data) did not contain Kinect nor respiration signals, the decision modules currently use only the acoustic laughter detection and acoustic laughter intensity. The other input components (in yellow on Figure 1) are nevertheless integrated in the system architecture and their data is recorded in order to train the decision modules with these additional signals in the future.

VI. INPUT COMPONENTS

To work properly, our system must be able to capture sufficient information about the user, coming from different

[2]http://www.semaine-project.eu/

[3]http://www.mindmakers.org/projects/bml-1-0/wiki/Wiki?version=10

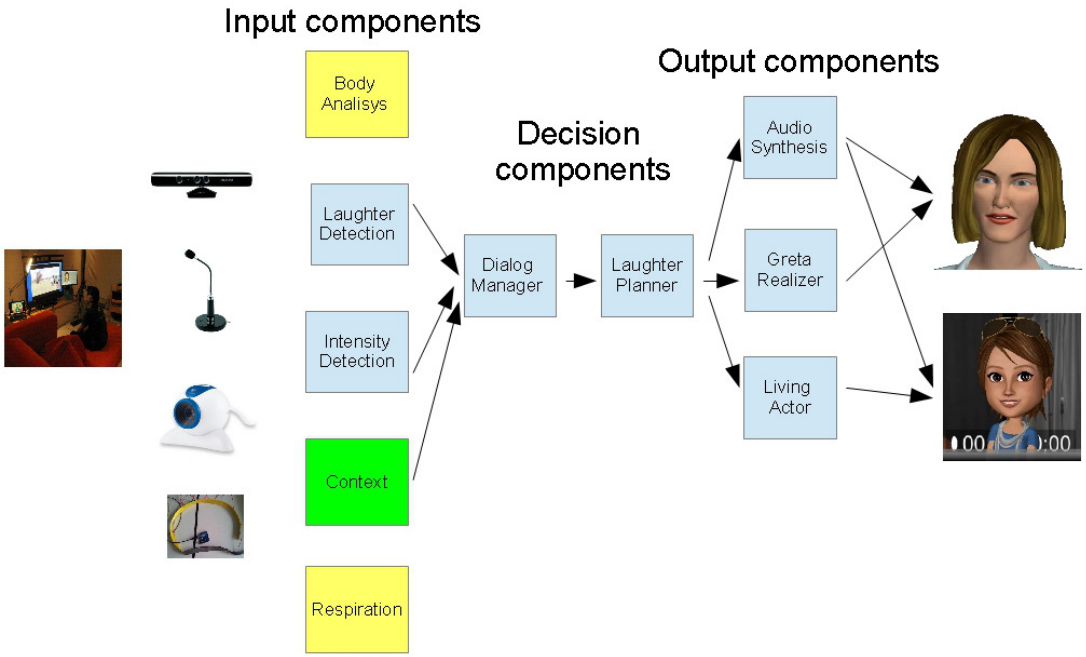


Fig. 1. Overall architecture of Laugh Machine

modalities such as sound, visual tracking, and chest movement. To facilitate the multimodal data processing and the synchronisation between the different signals, we have used the Social Signal Interpretation (SSI) [23] software developed at the University of Augsburg. This software will be presented first in this section, then we will present the different analysis components that have been developed: audiovisual laughter detection, laughter intensity estimation, respiration signal acquisition and body movement analysis. All these components have been plugged directly in SSI, except the body motion analysis, due to a problem of sharing the Kinect data in real-time.

A. SSI

The desired recognition component has to be equipped with certain sensory to capture multimodal signals. First, the raw sensor data is collected, synchronized and buffered for further processing. Then the individual streams are filtered, *e.g.* to remove noise, and transformed into a compact representation by extracting a set of feature values from the time- and frequency space. The in this way parameterized signal can be classified by either comparing it to some threshold or applying a more sophisticated classification scheme. The latter usually requires a training phase where the classifier is tuned using pre-annotated sample data. The collection of training data is thus another task of the recognition component. Often, an activity detection is required in the first place in order to identify interesting segments, which are subject to a deeper analysis. Finally, a meaningful interpretation of the detected events is only possible at the background of past events and events from other modalities. For instance, detecting several laughter events within a short time frame increases the probability that the user is in fact laughing. On the other hand, if we detect that the user is talking right now we would decrease the confidence for a detected smile. The different tasks the recognition component is involved with are visualized in Figure 2.

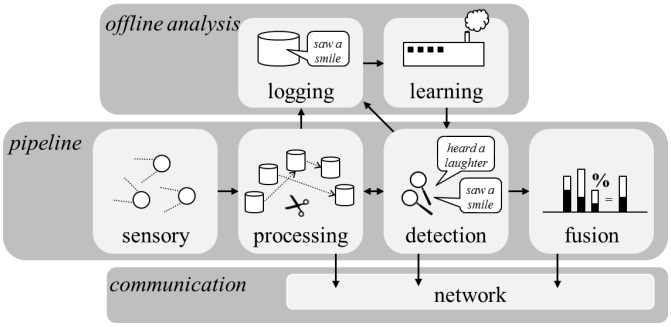


Fig. 2. Scheme of the laughter recognition component implemented with the Social Signal Interpretation (SSI) framework. Its central part consists of a recognition pipeline that processes the raw sensory input in real-time. If an interesting event is detected it is classified and fused with previous events and those of other modalities. The final decision can be shared through the network with external components. In order to train the recognition components a logging mechanism is incorporated in order to capture processed signals and add manual annotation. In an offline learning step the recognition components can now be tuned to improve accuracy.

The Social Signal Interpretation (SSI) software [23] developed at Augsburg University suits all mentioned tasks and was therefore used as a general framework to implement the recognition component. SSI provides wrappers for a large range of commercial sensors, such as web/dv cameras and multi-channel ASIO audio devices, as well as the Nintendo Wii remote control, Microsoft Kinect and various physiological sensors like NeXus, ProComp, AliveHeartMonitor, IOM or Emotiv. A patch-based architecture allows a developer to quickly construct pipelines to simultaneously manipulate the

raw signals captured by multiple devices, where the length of the processing window can be adjusted for each modality individually. Many common filter algorithms, such as moving and sliding average, Butterworth, Chebyshev, Elliptic, etc. as well as, derivative and integral filters are part of the core system and can be easily combined with a range of low-level features such as Fourier coefficients, intensity, cepstra, spectrogram, or pitch, as well as, more than 100 functionals, such as crossings, extremes, moments, regression, percentiles, etc. However, a plug-in system encourages developers to extend the core functions with whatever algorithm is required. A peak detection component is included, too, which can be applied to any continuous signal in order to detect segments above a certain activity level. If an event is detected it can be classified using one of various classification models such as K-Nearest Neighbor (KNN), Linear Discriminant Analysis (LDA), Support Vector Machines (SVM) or Hidden Markov Models (HMM). Tools for training and evaluation are available and can be combined with several feature selection algorithms (*e.g.*, SFS) and over-sampling techniques (*e.g.*, SMOTE [24]) for boosting under represented classes are available, too. Finally, classified events can be fused over time using vector-based event fusion. SSI offers a XML interface to put the different components to a single pipeline and keep control of important parameters.

In the Laugh Machine project, SSI was used for body and face tracking as well as audio and respiration recording. To have access to the new features provided in the latest Microsoft Kinect SDK, the Kinect wrapper in SSI was revised and updated accordingly. To access to the stretch values measured by the respiration sensor a new sensor wrapper was written using a serial connection.

After finishing the integration of the sensor devices, a recording pipeline was set up to record a training corpus for tuning the final recognition system (the Augsburg scenario recordings presented in Section IV-D). The pipeline also includes a playback component that allows replay of a video file to the user in order to induce laughter. This feature was used to drive the stimulus video directly from SSI. Since the video playback is then synchronized with the recorded signals, it is possible to relate captured laughter bouts to a certain stimuli in the video. The same pipeline was later used in our experiments. It is illustrated in Figure 3, which presents the Laugh Machine architecture from the point of view of SSI. The following sections present components that have been integrated into SSI: laughter detection, laughter intensity estimation and respiration signal acquisition.

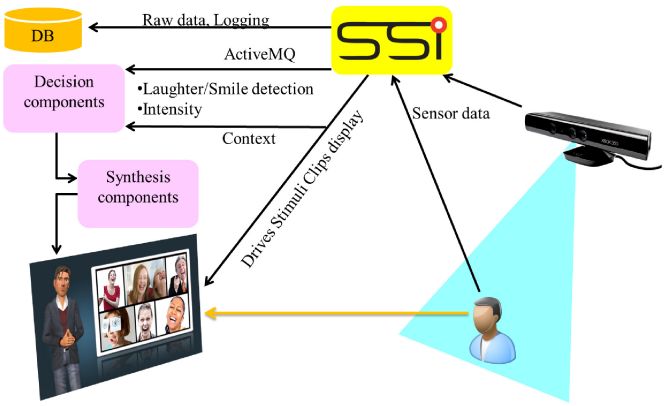


Fig. 3. SSI roles in the Laugh Machine system. While the user is watching funny video clips his or her non-verbal behavior is analyzed by a recognition component developed with SSI. If a laughter event is detected this information is shared to the behavior model, which controls the avatar engine. According to the input the avatar is now able to respond in an appropriate way, *e.g.*, join the user's laughter bout.

B. Laughter detection

Starting from the literature one can find several studies dealing with the detection of laughter from speech (*e.g.*, [1]–[3], see Section II-A). Most of them are pure offline studies and in part the explored feature types and classification methods vary largely. This circumstance makes it difficult to decide from scratch, which feature set and classifier would be the best choice for an online laughter detector. Hence, it was decided to run a fair comparison of the suggested methods in a large scale study. To this end, the SEMAINE database has been used and annotations of 19 files containing laughter were manually edited (as explained in Section IV-A).

Based on the edited annotations, for each second (a number commonly found in literature) it was decided whether the segment includes only silence (1906 samples), pure speech (5328), pure laughter (370), or both, speech and laughter (261). Samples were then equally distributed in a training and test set, while it was ensured that samples of the same user would not occur in both sets. To have an equal number of samples for each class, underrepresented classes were oversampled in the training set using SMOTE. After some preliminary tests it was decided to leave out silence, as it can be easily differed from speech and laughter using activity detection. It was also decided to leave out samples including both speech and laughter, as the goal of the experiment was to find features that best discriminate the two classes.

After setting up the database, large parts of the openSMILE (Speech & Music Interpretation by Large-space Extraction) feature extraction toolkit developed at the Technical University Munich (TUM) [25] were integrated into SSI. OpenSMILE is an open source state-of-the-art implementation of common audio features for speech and music processing. An important feature is its capability of on-line incremental processing, which makes it possible to run even complex and time-consuming algorithms, such as pitch extraction, in real-time. Based on the findings of earlier studies, the following speech-related low-level features were selected as most promising candidates: Intensity, MFCCs, Pitch, PLPs. On these the following 11 groups of functionals were tested: Zero-Crossings, DCT (Direct Cosine Transform) Coefficients, Segments, Times, Extremes, Means, Onsets, Peaks, Percentiles, Linear and Quadratic Regression, and Moments. Regarding classification, four well known methods were chosen: Naive Bayes (NB), Gaussian Mixture Models (GMM), Hidden Markov Models (HMM) and Support Vector Machines (SVM). Finally, the frame size at which low-level features are extracted was also altered.

A large scale experiment was then conducted. First, each of the 11 groups of functionals was tested independently with each of the four low-level feature types. In case of MFFCs also the number of coefficients was altered and higher-order derivatives (up to 4) were added. Results suggest that most reliable results are achieved using Intensity and MFCCs, while adding Pitch and PLP features did not improve results on the studied corpus. Among the functionals, Regression, Moments, Peaks, Crossings, Means and Segments are considered to carry most distinctive information. Regarding classification, SVM with a linear kernel clearly outperformed all other tested recognition methods. In terms of operation size accuracy was highest at a frame rate of 10ms with 2/3 of overlap. In the best case an overall accuracy of 88.2% at an unweighted average recall of 91.2% was obtained.

The developed laughter detection framework was then tuned to the specific Laugh Machine scenario and input components (*i.e.*, the audio is recorded by the Kinect), thanks to the Augsburg scenario recordings (see Section IV-D). The annotations of the audio tracks were used to re-train the laughter detector described above, with the features extracted in the Laugh Machine scenario conditions. The obtained laughter model was finally combined with a silent detection to filter out silent frames in the first place and classifying all remaining frames into laughter or noise. The frame size was set to 1 second with an overlap of 0.8 second, *i.e.* a new classification is received every 0.2 second. The annotations of the video tracks are meant for training an additional smile detector in the future. Same counts for the respiration signal (see Section VI-D), which in future will serve as a third input channel to the laughter detector.

C. Laughter intensity

Knowing the intensity of incoming laughs is important information to determine the appropriate behavior of the virtual agent.

In [26], naive participants have been asked to rate the intensity of laughs from the AVLaughterCycle database [19] on a scale from 1 (very low intensity) to 5 (very high intensity). One global intensity value had to be assigned to each laugh. Audiovisual features that correlate with these perceived global intensity have then be investigated.

Here, we wanted not only to estimate the global laughter intensity, after the laugh has finished, but to measure in real-time the instantaneous intensity. As a first step, only the audio modality was included. 49 acoustic laughs, produced by 3 subjects of the AVLaughterCycle database and distributed over the ranges of annotated global intensity values, have been continuously annotated in intensity by one labeler. Acoustic features have been extracted with the objective to predict the continuous intensity curves.

Figure 4 displays the manual intensity curve for one laugh, together with the automatic intensity prediction obtained from two acoustic features: loudness and pitch. The intensity curve is obtained by a linear combination between the maximum pitch and the maximum loudness values over a sliding 200ms window, followed by median filtering to smooth the curve. The overall trend is followed, even though there are differences, mostly at the edge of the manually spotted bursts, and the manual curve is smoother than the automatic one. Furthermore, the overall laughter intensity can be extracted from the continuous annotation curve: correlation coefficients between the median intensity scored by users and the intensity predicted from acoustic features are over 0.7 for 21 out of 23 subjects[4].

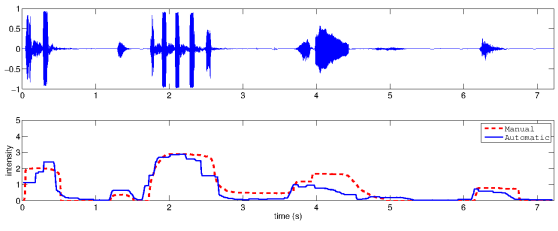


Fig. 4. Example of laughter continuous intensity curve. Top: waveform; Bottom: manual and automatic intensity curves.)

During the eNTERFACE workshop, work has been done to improve the computation of the continuous intensity curve. Indeed, the linear combination is able to capture trends for one subject (which laugh or laugh segment is more intense than another one), but the outputted values fall in different ranges from one subject to another. Classification with Weka [27] has been investigated to overcome this problem. First, neural networks have been trained in Weka to predict the continuous intensity curve from acoustic features (MFCCs and spectral flatness). The correlation with the manually annotated curves was over 0.8, using a "leave-one-subject-out" scheme for testing. Second, other neural networks have been used to compute the global laughter intensity from the predicted continuous intensity. To keep the number of features constants, 5 functionals (max, std, range, mean, sum) of the continuous intensity have been used as inputs. The results again show a good correlation between the predicted global intensity and the one rated by naive participants, in this case with similar values for all the subjects of the AVLaughterCycle database.

However, the speaker-independent intensity detection with Weka could not be integrated in the full LaughMachine system yet. Only the linear combination has been used in our experiments. Further work to improve laughter intensity prediction include the extension of the feature set to visual features, the integration of the Weka classification within the Laugh Machine framework and possibly the adaptation of the functions to the user.

D. Respiration

The production of audible laughter is, in essence, a respiratory act since it requires the exhalation of air to produce distinctive laughter sounds ("Ha") or less obvious sigh- or hiss-like verbalizations. The respiratory patterns of laughter have been extensively researched as Ruch & Ekman [21] summarize. A distinctive respiration pattern has emerged of

[4]The 24th subject of the AVLC corpus only laughed 4 times and all these laugsh were rated witht he same global intensity, which prevents us from computing correlations for this subject

a rapid exhalation followed by a period of smaller exhalations at close-to-minimum lung volume. This pattern is reflected by changes in the volume of the thoracic and abdominal cavities, which rapidly decrease to reach a minimum value within approximately 1 s [28]. These volumetric changes can be seen through the simpler measure of thoracic circumference, noted almost a century ago by Feleky [29]. In order to capture these changes, we constructed a respiration sensor based on the design of commercially available sensors: the active component is a length of extensible conductive fabric within an otherwise inextensible band that is fitted around the upper thorax. Expansions and contraction of the thorax change the length of the conductive fabric causing changes in its resistance. These changes in resistance are used to modulate an output voltage that is monitored by the Arduino prototyping platform[5]. Custom-written code on the Arduino converts the voltage to a 1-byte serial signal, linear with respect to actual circumference, which is passed to a PC over a USB connection at a rate of approximately 120Hz.

Automatic detection of laughter from respiratory actions has previously been investigated using electromyography (EMG). Fukushima et al. analyzed the frequency characteristics of diaphragmatic muscle activity to distinguish laughter, which contained a large high-frequency component, from rest, which contained mostly low-frequency components [15]. We exploited the predictable respiration pattern of laughter to use simpler techniques that do not rely on computationally demanding frequency decomposition. We identified laughter onset through the appearance of 3 respiration events (see Figure 5):

1) A sharp change in current respiration state (inhalation, pause, standard exhalation) to rapid exhalation.
2) A period of rapid exhalation resulting in rapid decrease in lung volume.
3) A period of very low lung volume

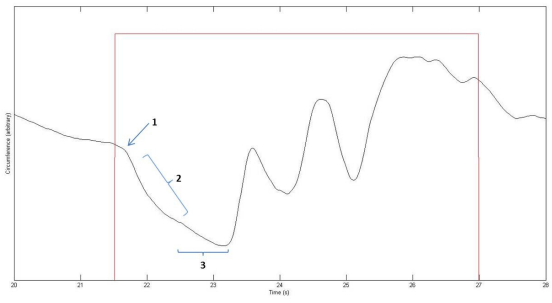


Fig. 5. Example of thoracic circumference, with laughter episode marked in red, and notable features of laughter initiation. Feature 1 - a sharp change in current respiration state to rapid exhalation; feature 2 - a period of rapid exhalation; feature 3 - a period of very low lung volume.

These appear as distinctive events in the thoracic circumference measure and its derivatives:

1) A negative spike in the second derivative of thoracic circumference.
2) A negative period in the first derivative of thoracic circumference.
3) A period of very low thoracic circumference.

These were identified by calculating a running mean (λ_f) and standard deviation (σ_f) for each measure. A running threshold (T_f) for each measure was calculated as:

$$T_f = \lambda_f - \alpha_f \sigma_f \quad (1)$$

where α_f is a coefficient for that measure, empirically determined to optimise the sensitivity/specificity trade-off. Each feature was determined to be present if the value of the measure fell below the threshold at that sample. Laughter onset was identified by the presence of all three features in the relevant order (1 before 2 before 3) in a sliding window of approximately 1 s. This approach restricts the number of parameters to 3 ($\alpha_{1\text{—}3}$) but does introduce lag necessary for calculating valid derivatives from potentially noisy data. It also requires a period for the running means and standard deviations, and so the running thresholds, to stabilise. This process would be jeopardised by the presence of large, rapid respiratory event such as coughs and sneezes. We were unable to integrate these rules into the LaughMachine system due to technical errors. Future recordings on the LaughMachine platform, incorporating the respiration data, will allow optimisation of these rules and the fusion of respiration data with other modalities for real-time laughter/non-laughter discrimination.

E. Body analysis

The EyesWeb XMI platform is a modular system that allows both expert (e.g., researchers in computer engineering) and non-expert users (e.g., artists) to create multimodal installations in a visual way [30]. The platform provides modules, called blocks, that can be assembled intuitively (*i.e.*, by operating only with mouse) to create programs, called patches, that exploit system's resources such as multimodal files, webcams, sound cards, multiple displays and so on. The body analysis input component consists of an EyesWeb XMI patch performing analysis of the user's body movements in realtime. The computation performed by the patch can be split into a sequence of distinct steps, described in the following subsections.

1) Shoulder tracking: The task of the body analysis module is to track the user's shoulders and perform some computation on the variation of their position in realtime. In order to do that we could provide the Kinect shoulders' data extracted by SSI (see Section VI-B) as input to our component. However, we observed that the shoulders' position extracted by Kinect do not consistently follow the user's real shoulder movement: in the Kinect skeleton, shoulders' position is determined by performing some statistical algorithm on the user's silhouette and depth map and usually this computation can not track subtle shoulders' movement, for example, small upward/downward movements. To overcome this limitation we fixed two markers on the user's body: two small and lightweight green polystyrene spheres have been fixed on the user's clothes just over the user's shoulders. The EyesWeb patch separates the green channel of the input video signal

[5]http://www.arduino.cc/

to isolate the position on the video frame of the two spheres. Then a tracking algorithm is performed to follow the motion of the sphere frame by frame, as shown in Figure 6.

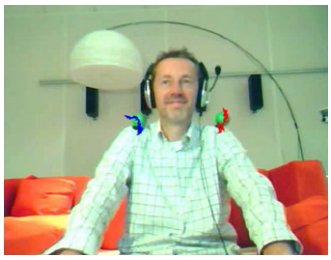


Fig. 6. Two green spheres placed on the user's shoulders are tracked in realtime (red and blue trajectories)

The position of each user's shoulder is associated to the barycenter of each sphere, which can be computed in two ways. The first consists in the computation of the graphical barycenter of each sphere, that is, the mean of the pixels of each sphere's silhouette is computed. The second option includes some additional steps: after computing the barycenter like in the first case, we consider a square region around it and we apply a Lukas-Kanade [31] algorithm to this area. The result is a set of 3 points on which we compute the mean: the resulting point is taken as the position of the shoulder.

2) Correlation: Correlation ρ is computed as the Pearson correlation coefficient between the vertical position of the user's left shoulder and the vertical position of the user's right shoulder. Vertical positions are approximated by the y coordinate of each shoulder's barycenter extracted as mentioned above.

3) Kinetic energy: It is computed from the speed of user's shoulders and their percentage mass as referred by [32] :

$E = \frac{1}{2}(m_1 v_1 + m_2 v_2)$

4) Periodicity: Kinetic energy is serialized in a sliding window time-series having a fixed length. Periodicity is then computed on such time-series, using Periodicity Transforms [33]. The input data is decomposed into a sum of its periodic components by projecting data onto periodic subspaces. Periodicity Transforms also provide a measure of the relative contribution of each periodic signal to the original one. Among many algorithms for computing Periodicity Transforms, we chose mbest. It determines the m periodic components that, subtracted from the original signal, minimize residual energy. With respect to the other algorithms, it also provides a better accuracy and does not need the definition of a threshold.

5) Body Laughter Index: Body Laughter Index (BLI) stems from the combination of the averages of shoulders' correlation and kinetic energy, integrated with the Periodicity Index. Such averages are computed over a fixed range of frames. However such a range could be automatically determined by applying a motion segmentation algorithm on the video source. A weighted sum of the mean correlation of shoulders' movement and of the mean kinetic energy is carried out as follows:

$BLI = \alpha\bar{\rho} + \beta\bar{E}$

As reported in [21], rhythmical patterns produced during laughter usually have frequencies around 5 Hz. In order to take into account such rhythmical patterns, the Periodicity Index is used. In particular, the computed BLI value is acknowledged only if the mean Periodicity Index belongs to the arbitrary range $[\frac{fps}{8}, \frac{fps}{2}]$, where fps is the input video frame rate (number of frames per second).

6) ActiveMQ: The EyesWeb XMI platform can be expanded to implement new functionalities that could be included into new sets of programming modules (blocks). To allow the communication between the body analysis patch and the other components (e.g., the SSI audio and face analysis component) we implemented two new blocks: the ActiveMQ receiver and the ActiveMQ sender. Body analysis component sends two types of data using the ActiveMQ message format described in Section V: data messages and clock messages. Data messages contain tuples representing the values of the user's shoulders movement features presented above. Clock messages contain the system clock of the machine on which the EyesWeb XMI platform is running. They are sent to the ActiveMQ server on which all the other components are registered. So, the local clock of all the components (audio and face analysis, dialogue generation and so on) is constantly updated with the same value and synchronization between the different component can be assured. In the future we aim to exploit the synchronization features embedded in the SEMAINE platform, that is implemented as a layer of the ActiveMQ communication protocol.

VII. Dialog manager

The laughter-enabled dialogue management module aims at deciding, given the information from the input components (*i.e.*, laughter likelihoods and intensity from multimodal features) as well as contextual information (*i.e.*, the funniness of the stimulus), when and how to laugh so as to generate a natural interaction with human users. In this purpose, the dialogue management task is seen as a sequential decision making process meaning that the behavior is not only influenced by the current context but also by the history of the dialogue. This is a main difference comparing with the other interactive systems such as SEMAINE. The optimal sequence of decisions is learned from actual human-human or human-computer interaction data and is not rule-based or handcrafted which is another difference with the SEMAINE system.

The decision of whether and how to laugh must be taken at each time frame. For the eNTERFACE workshop a time frame lasts $\Delta t = 200$ms. The input I received by the Dialog manager at each time frame is a vector ($I \in [0,1]^k$: each feature has been normalized) where k is the number of chosen multimodal features. The output O produced at each time frame is a vector ($O \in [0,1] \times [0, \text{time}_{\text{max}}]$) where the first dimension codes the laughter intensity and the second dimension codes the duration of the laugh.

The method used to build the decision rule during the eNTERFACE workshop is a supervised learning method. A supervised learning method is able via a training data set $D = \{x_i, y_i\}_{1 \leq i \leq J}$ ($\{x_i\}_{1 \leq i \leq J}$ are the inputs which belong to the set X, $\{y_i\}_{1 \leq i \leq J}$ are the labels which belong to the set Y and $J \in \mathbb{N}^*$) to build a decision rule π. The decision rule π is a function from X to Y that generalizes the relation between

the inputs x_i and the labels y_i of the training data set. There are two different types of supervised methods: Classification when the number of outputs is finite and Regression when the number of outputs is infinite.

A. Training of the Dialog manager

To apply a supervised learning method to our dialog manager, we need a training data set specific to our scenario (see Section III). The Belfast interaction dyads (see Section IV-C) was recorded to this purpose. Let us name the two interacting participants $P1$ and $P2$, respectively recorded on tracks $T1$ and $T2$. We recall that the participants watch simultaneously the same stiumulus video, and can also see (and hear) each other on the display screen: $P2$ is viewable by $P1$ and is considered as playing the role of the virtual agent. The length of a recording is $H = K\Delta t$. Thus, on $T1$ we have the inputs (*i.e.*, laughter likelihoods and intensity from multimodal features of $P1$) $\{I_i\}_{1\leq i\leq K}$ of the Dialog manager and on $T2$ the corresponding outputs $\{O_i\}_{1\leq i\leq K}$ which are the intensities and durations of the laughs of $P2$.

The aim of the supervised method is to find a decision rule such that the virtual agent will be able to imitate $P2$. Before applying a supervised method, we decided to cluster the inputs with N clusters (via a k-means method) and to cluster the outputs with M clusters (via a Gaussian Mixture Model, or GMM, method). k-means clustering is a method of cluster analysis which aims to partition $n \in \mathbb{N}^*$ observations into $0 \leq k \leq n$ clusters in which each observation belongs to the cluster with the nearest mean. This results in a partitioning of the data space into Voronoi cells. GMM clustering is a method of cluster analysis where each cluster can be parameterized by a Gaussian distribution. The choice of the GMM method for the output clustering is explained in Section VII-B.

Thanks to clustering, the input data becomes the input clustered data $\{I_i^C\}_{1\leq i\leq K}$ with $I_i^C \in \{1,\dots,N\}$ and the output data becomes the output clustered data $\{O_i^C\}_{1\leq i\leq K}$ with $O_i^C \in \{1,\dots,M\}$. Clustering the inputs allows to have a finite decision rule which means that the decision rule can be represented by a finite vector. Clustering the outputs allows using a classification method such as the k-nearest neighbors (k-nn) instead of a regression method which is more difficult to implement.

Finally the supervised method used on the clustered data $\{I_i^C, O_i^C\}_{1\leq i\leq K}$ is a k-nearest-neighbor method which gives us the decision rule π which is a function from $\{1,\dots,N\}$ to $\{1,\dots,M\}$. k-nn is a method for classifying objects based on closest training examples: the object is assigned to the most common label amongst its k nearest neighbors. Figure 7 represents the training phase of the Dialog manager needed to obtain the decision rule π.

B. Using the Dialog manager

The decision rule π obtained by the classification method on $\{I_i^C, O_i^C\}_{1\leq i\leq K}$ is a function from $\{1,\dots,N\}$ to $\{1,\dots,M\}$: it takes an input cluster and it gives an output cluster. However, our dialog manager must be able to take an input $I \in [0,1]^k$ and give an output $O \in [0,1] \times [0, \text{time}_{\max}]$.

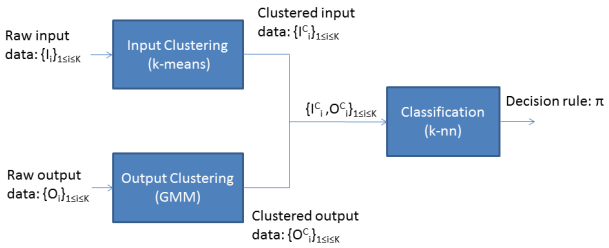


Fig. 7. Dialog Manager training

So, first we need to assign the input $I \in [0,1]^k$ to the corresponding input cluster $I^C \in \{1,\dots,N\}$. To do that, we choose the cluster for which the mean is the closest to $I \in [0,1]^k$:

$$I_C = \underset{1\leq i\leq N}{\operatorname{argmin}} \|I - \mu_i^I\|_2^2, \tag{2}$$

where μ_i^I is the mean of the input cluster $i \in \{1,\dots,N\}$ and $\|\|_2$ is the euclidean norm. This operation is called the input cluster choice. Second, to be able to generate O from the selected output cluster $l \in \{1,\dots,M\}$ the question is: which element of the output cluster l must we choose in order to correspond to the data $\{O_i\}_{1\leq i\leq K}$? This is why we use a GMM method for clustering the outputs: each cluster l can be seen, in the 2-dimensional intensity-duration plane, as a Gaussian of law $\mathcal{N}(\mu_l^O, \Sigma_l^O)$, where μ_l^O is the mean of the output cluster l and Σ_l^O is the covariance matrix of the output cluster l. Therefore, to obtain an output, it is sufficient to sample an element O of law $\mathcal{N}(\mu_l^O, \Sigma_l^O)$. This operation is called the output generation.

Let us summarize the functioning of the Dialog manager (see also Figure 8): we receive the input I, we associate this input to its corresponding input cluster $I^C \in \{1,\dots,N\}$, then the decision rule π gives the output cluster $\pi(I^C) \in \{1,\dots,M\}$, finally the output O is chosen in the output cluster $\pi(I^C) \in \{1,\dots,M\}$ via the output generation.

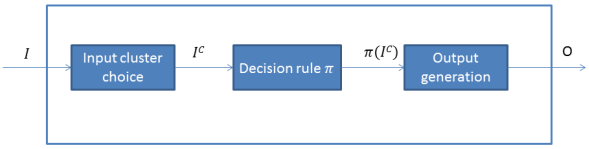


Fig. 8. Dialog Manager functioning

C. Laughter Planner

In the Laugh Machine architecture, the dialog manager is followed by the Laughter Planner, which is adapting the outputs of the dialog manager to the constraints (instruction format, avoid conflicting information, etc.) of the synthesis modules. While it technically is a decision component, the explanations about the Laughter Planner are included in the visual synthesis section (Section VIII-B).

VIII. Audiovisual Laughter synthesis

A. Acoustic laughter synthesis

Given 1) the lack of naturalness resulting from previous attempts to laughter acoustic synthesis, 2) the need for high level control of the laugh synthesizer and 3) the good performance achieved with Hidden Markov Model (HMM) based speech synthesis [34], we decided to investigate the potential of this technique for acoustic laughter synthesis. We opted for the HMM-based Speech Synthesis System (HTS) [35], as it is free and widely used in speech synthesis and research.

Explaining the details of speech synthesis with HMMs or HTS going beyond the scope of this project report, we will here only describe the major modifications that have been brought to adapt our laughter data to HTS and vice-versa, adapting functions or parameters of the HTS demo (provided with the HTS toolbox) to improve the quality of laughter synthesis. Readers who would like to know more about HTS are encouraged to consult the publications listed on the HTS webpage (http://hts.sp.nitech.ac.jp/?Publications), and in particular [34] for an overview or Yoshimura's Phd Thesis [36] for more detailed explanations.

The following paragraphs respectively focus on the selection of the training data, the modifications implemented in the HTS demo and, finally, the resulting process for acoustic laughter synthesis.

1) Selection and adaptation of acoustic data:

HMM-based acoustic synthesis requires a large quantity of data: statistical models for each unit (in speech: phonemes) can only be accurately estimated if there are numerous training examples. Furthermore, the data should be labeled (*i.e.*, a phonetic transcription must be provided) and come from a single person, whose voice is going to be modeled. HMM-based speech synthesis is usually trained with hours of speech.

It is difficult to obtain such large quantities of spontaneous laughter data. The only laughter database including phonetic transcriptions is the AVLaughterCycle database [19], [20], which contains in total 1 hour of laughter from 24 subjects. We decided to use that database for our acoustic laughter synthesis.

To fully exploit the potential of HTS, the phonetic annotations of the AVLaughterCycle database have been extended to syllables. Indeed, HTS is able to distinguish contexts that lead to different acoustic realizations of a single phoneme (and on the other hand, HTS groups the contexts that yield acoustically similar realizations of a phoneme). In speech, the context of a phoneme is defined not only with the surrounding phonemes, but also with prosodic information such as the position of the phoneme within the syllable, the number of phonemes in the previous, current and following syllables; the number of syllables in the previous, current and following words; the number of words in the phrase; etc. Except from the surrounding phones[6], such contextual information was not available in the AVLC annotations, as there was no annotation of the laughs in terms of syllables or words. It was decided to add a syllabic annotation of the data to provide the biggest possible contextual information. There is no clear definition of laughter syllables, and the practical definition that has been used for the syllabic annotation was to consider one syllable as a set of phones that was acoustically perceived as forming one block (or burst), usually containing one single vowel (but not always, as laughter can take different structures from speech). Since the syllabic annotation is time-consuming, it was decided to do it only for the subjects who laughed the most in the AVLaughterCycle database: subjects 5, 6, 14, 18 and 20. These subjects laugh around 5 minutes each, which is already far from the hours of training data used in speech synthesis, and it seemed they represent the best hopes for good quality laughter synthesis. The HTS contextual information was then formed by assimilating a full laughter episode to a speech sentence and laughter exhalation and inhalation segments to words.

In addition, due to the limited available data, the phonetic labels have been grouped in 8 broad phonetic classes—namely: fricatives, plosives, vowels, hum-like (including nasal consonants), glottal stops, nareal fricatives (noisy respiration airflow going through the nasal cavities), cackles (very short vowel similar to hiccup sound) and silence—instead of the 200 phones annotated in the AVLaughterCycle database [20]. Indeed, most of these phones had very few examples for each speaker, and hence could not be accurately modeled. Grouping acoustically similar phones enables to obtain better models, at the cost of reduced acoustic variability (*e.g.*, all the vowels are grouped in an average model that is close to 'a', and we loose the possibility to generate the few 'o's in the database).

An example of the resulting phonetic transcription is presented in Figure 9.

Finally, the laughs from the AVLaughterCycle database have been processed to reduce background noise and remove saturations.

2) Modifications of the HTS demo process:

Several minor modifications have been applied to HTS. Some of them are simple parameter variations compared to the standard values used in speech (and in the HTS demo). For example, the boundaries for fundamental frequency estimation have been extended (the values have been manually determined for each subject), the threshold for pruning decision trees has been increased, etc. In addition the list of questions available to decision trees has been extended, considering the new contextual information available for laughter.

More important, two standard HTS algorithms have been replaced by more efficient methods. First, the standard Dirac pulse train for voiced excitation has been replaced by the DSM model [37], which better fits the human vocal excitation shapes and reduces the buzziness of the synthesized voice. Second, the standard vocal tract and fundamental frequency estimation algorithms provided by HTS have been replaced by the STRAIGHT method [38], which is known in speech processing to provide better estimations.

3) Synthesis process:

With the explained modifications to the AVLaughterCycle database and the HTS demo, we were able to train laughter synthesis models, with which we can produce acoustic laughs

[6]Since the phonological notion of "phoneme" is not clearly defined for laughter; we prefer to use the word "phone" for the acoustic units found in our laughter database.

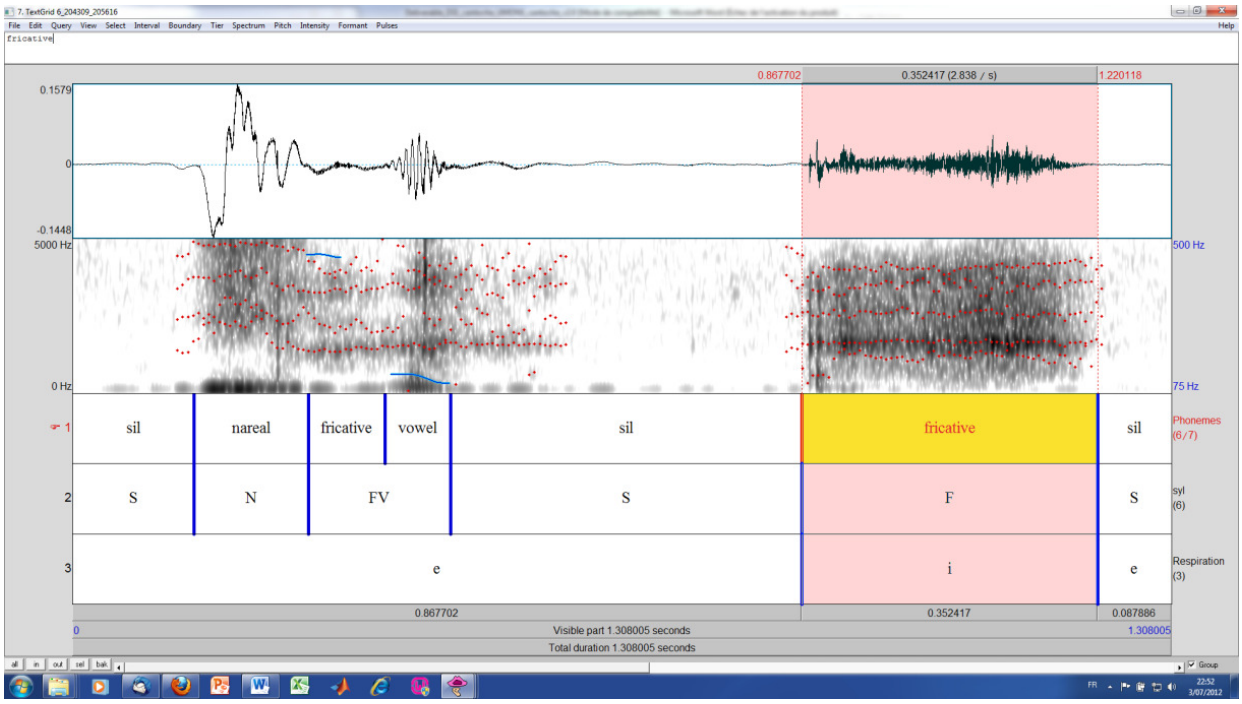


Fig. 9. Laughter phonetic and syllabic annotation: from top to bottom: a) waveform b) spectrogram c) phonetic annotation (using the 8 broad classes) d) syllable annotation e) respiration phases (inhalation or exhalation)

when giving an acoustic laughter transcription as input. It is worth noting that there is currently no module to generate such laughter phonetic transcriptions from high-level instructions (*e.g.*, a type of laughter, its duration and its intensity). We are thus constrained to play existing laughter transcriptions. Additionally, we noticed that the synthesis quality drops if we want to synthesize a phonetic transcription from speaker A with the models trained on voice B. In consequence, we currently stick to re-synthesizing laughs from one speaker, using both the phonetic transcription and the models trained from the same subject.

A perceptive evaluation study still has to be carried out. Nevertheless, the first qualitative tests are promising. The modifications explained in the previous paragraphs largely improved the quality of the laughter synthesis. There remain some laughs or sounds that are not properly synthesized, possibly due to the limited training data. Future works will investigate this issue as well as the possibility to generate new laughter phonetic transcriptions (or modify existing ones) that can be synthesized properly. Nevertheless, at the end of this project, we are able to synthesize a decent number of good quality laughs for the best voices coming from the AVLaughterCycle database.

B. Visual laughter Synthesis

Two different virtual agents and four different approaches were used for the visual synthesis. The visual synthesis component is composed of a Laughter Planner and 2 Realizers and Players (see Figure 10).

The Laughter Planner receives from the dialog manager the information about the appropriate laugh reaction through the ActiveMQ/SEMAINE architecture (see Section VII). Next it chooses one laugh episode from the library of predefined laugh samples and generates the appropriate BML command that is sent through ActiveMQ/SEMAINE to one out of two realizers available in the project: Living Actor or Greta Realizer.

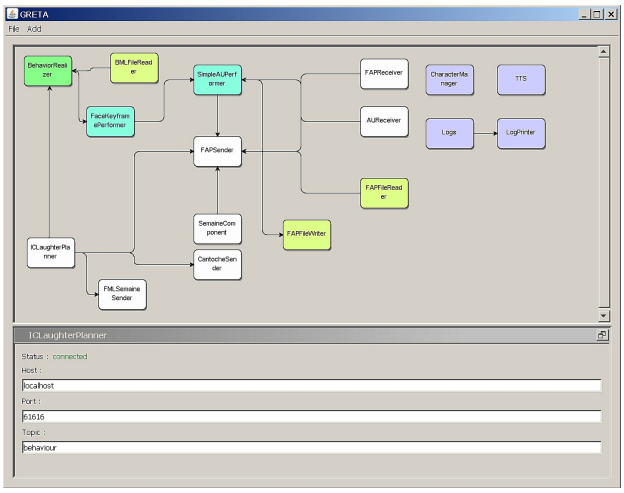


Fig. 10. Visual Synthesis Component Pipeline

On Figure 10 we present the detailed processing pipeline of our visual synthesis component. The Laughter Planner is connected to the Greta Behavior Realizer and the Cantoche Sender. The latter is responsible for the communication with the Living Actor component (see Section VIII-B4). Both Behavior Realizer and Cantoche Sender receive the same BML message. As these realizers use completely different methods for controlling the animation (Greta can be controlled by high-

level facial behavior description in FACS and low-level facial animation parameterization (MPEG-4/FAPs) while Living Actor plays predefined animations) we use realizer-specific extensions of BML to assure that the animations played with different agents are similar. If necessary, the Laughter Planner can also send commands in a high-level language called FML (FMLSemaineSender box) or control facial animations at very low level by specifying the values of facial animation parameters (FAPs) (FAPSender box). Independently of which of these pipelines is used the final animation is described using low level facial animation parameters (FAPs) and is sent through ActiveMQ/SEMAINE to the visualization module (FAPsender box). At the moment we use the Player from the SEMAINE Project. Four characters are included in this Player (2 male, 2 females) but for the purpose of the evaluation we used only one of them.

The Laughter Planner module can work in three different conditions, related to the three experimental scenarios: fixed speech condition (FSC), fixed laughter condition (FLC) and interactive laughter condition (ILC). In the first two conditions (FSC and FLC), the Laughter Planner receives the information about the context (time of funny event, see Section IX-C2) and it sends the agent verbal (FSC) or nonverbal (FLC) reaction pre-scripted in BML to be displayed to the user. The list of these behaviors was chosen manually.

In ILC condition the behavior of the agent is flexible as it is adapted to the participant and the context. The Laughter Planner receives the information on duration and intensity of laughter responses and using these values it chooses one laugh episode from the library that matches the best both values.

At the moment, the synthesis components do not allow for interruptions of the animation. Once it is chosen, the laugh episode has to be played until the end. During this period the Laughter Planner does not take into the account any new information coming from dialog manager. All the episodes start and end with a neutral expression. Thus they cannot be concatenated without passing through neutral face. Additionally the presynthesized audio wave file was synchronized with the animation.

Four different approaches were used in the project to prepare the lexicon of laughs: animation from the manual annotation of action units; animation from automatic facial movements detection; motion capture data driven; and manual animation. They are explained in the next subsections.

1) Animation from manual Action Units:

The Facial Action Coding System (FACS; [39]) is a comprehensive anatomically based system for measuring all visually discernible facial movement. It describes all distinguishable facial activity on the basis of 44 unique Action Units (AUs), as well as several categories for head and eye position movements and miscellaneous actions. Facial expressions of emotions are emotion events that comprise of a set of different AUs expressed simultaneously. Using FACS and viewing digital-recorded facial behavior at frame rate and in slow motion, certified FACS coders are able to distinguish and code all possible facial expressions. Utilizing this technique, a selection of twenty pre-recorded, laboratory stimulated, laughter events were coded. These codes were then used to model the facial behavior on the agent.

Four subjects interacting in same sex dyads watching the stimulus videos (see Section IV-C) were annotated by one certified FACS coder. Inter rater reliability was obtained by the additional coding of 50% of the videos by a second certified coder. The inter-rater reliability was sufficient (r = .80) and consent was obtained on events with disagreement. Furthermore, a selection of 20 laughter events from the AVLC laughter database [19] (subject 5) were coded by one certified coder.

The Greta agent is able to display any configuration of action units. For 3 characters (two females—Poppy and Prudence— and one male—Obadiah) single action units were defined and validated by certified FACS coders. A BML language implemented in Greta permits to control independently each action unit of the agent (its duration and intensity).

Furthermore, as a quality control, the animated AUs of the virtual agent was scrutinized by the FACS coders for a) anatomical appearance change accuracy, b) subtle differences and dominance rules relating to changes in the face when different intensity of facial expressions are produced.

During eNTERFACE we also developed a tool that automatically converts manual FACS annotation files to BML. Consequently any file containing manual annotation of action units can be easily displayed with the Greta agent.

2) Animation from Automatic Facial Movements detection: Greta uses Facial Animation Parameters (FAPs) to realize low level facial behavior. FAPs in Greta framework are represented as movements of MPEG-4 facial points compared to 'neutral' face. In order to estimate FAPs of natural facial expressions, we make use of an open-source face tracking tool—FaceTracker [40]—to track facial landmark localizations. It uses a Constrained Local Model (CLM) fitting approach that includes Regularized Landmark Mean-Shift (RLMS) optimization strategy. It can detect 66 facial landmark coordinates within real-time latency depending on system's configuration. Figure 11 shows an example of 2D and 3D landmark coordinates predicted by FaceTracker.

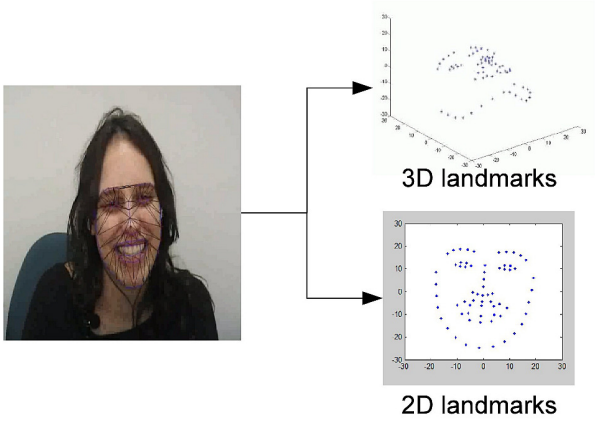


Fig. 11. Landmarks estimated by FaceTracker

Facial geometry is different for one and another. Therefore, it is difficult to estimate FAPs without neutral face calibration. To compute FAPs from facial landmarks, a neutral face model

is created with the help of 50 neutral faces of different persons. With the help of this model, FAPs are estimated as the distance between facial landmarks and neutral face landmarks. In case of user-specific FAP estimation in real-time scenario, the neutral face is estimated from a few seconds of video by explicitly requesting the user to be neutral. However, the better estimation of FAPs requires manual intervention for tweaking weights to map landmarks and FAPs, which is a down-side of this methodology. Figure 12 shows comparison of the locations between MPEG-4 FAP standard and the FaceTracker's landmark localizations.

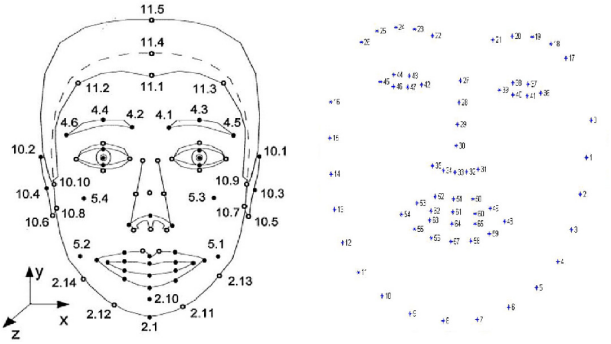


Fig. 12. (a) MPEG-4 FAP standard [left];(b) FaceTracker's landmark locations [right].

The landmark coordinates produced by the FaceTracker are observed as noisy due to the discontinuities and outliers in each facial point localization. Especially, the realized behavior is unnatural when we re-target the observed behavior onto Greta. In order to smooth the face tracking parameters, a temporal regression strategy is applied on individual landmarks by fitting 3rd order polynomial coefficients on a sliding window, where the sliding window size is 0.67 seconds (*i.e.*, 16 frames) and sliding rate is 0.17 seconds (*i.e.*, 4 frames). An example of the final animation can be seen on Figure 13.

3) Animation from Motion Capture Data:

The AVLC corpus (see Section IV-B) contains motion capture data of laugh episodes that has to be retargeted to the virtual model. The main problem in this kind of approaches consists in finding appropriate mappings for each participant's face geometry and different virtual models. Existing solutions are typically linear (*e.g.*, methods based on blendshape mapping) and do not take into account dynamical aspects of the facial motion itself. Recently Zeiler et al. [41] proposed to apply variants of Temporal Restricted Boltzmann Machines[7] (TRBM) to facial retargeting problem. TRBM are a family of models that permits tractable inference but allows complicated structures to be extracted from time series data. These models can encode a complex nonlinear mapping from the motion of one individual to another which captures facial geometry and dynamics of both source and target. In the original application [41] these models were trained on a dataset of facial motion capture data of two subjects, asked to perform a set of isolated facial movements based on FACS. The first subject had 313 markers (939 dimensions per frame) and the second subject had 332 markers (996 dimensions per frame). Interestingly there was no correspondence between marker sets.

We decided to use TRBM models for our project which involves retargeting from an individual to a virtual character. In our case, we take as input the AVLC mocap data and output the corresponding facial animation parameters (FAP) values. This task has two interesting aspects. First, the performance of these models was previously evaluated only on retargeting an isolated slow expression whereas our case involves transitions from laughing to some other expression (smile or idleness) as well as very fast movements. Second, we use less markers comparing to the original application. Our mocap data had only 27 markers on the face which is very sparse.

So far we used the AVLC data on one participant (number 5) as a source mocap data. We used two sequences, one of 250 frames and another one of 150 frames, to train this model. Target data (*i.e.*, facial animation parameters) for this training set was generated using the manual retargeting procedure explained in [13]. Both the input and output data vectors were reduced to 32 dimensions by retaining only their first 32 principal components. Since this model typically learns much better on scaled data (around [-1,1]), the data was then normalized to have zero mean and scaled by the average standard deviation of all the elements in the training set. Having trained the model, we used it to generate facial animation parameters values for 2 minutes long mocap data (2500 frames coming from the same participant). The first results are promising but more variability in the training set is needed to retarget more precisely different type of movements. It is important to notice that this procedure needs to be repeated for each virtual model (*e.g.*, Poppy, Prudence, Obadiah).

4) Manual Animation:

The Laugh Machine Living Actor module is composed of a real-time 3D rendering component using Living Actor technology and a communication component that constitutes the interface between the Living Actor agent and the ActiveMQ messaging system. Two virtual characters have been chosen for the first prototype: a girl and a boy, both with cartoonish style. Two types of laughter animations were created for each one by 3D computer graphics artists by visually matching the real person movies from the video database of interacting dyads (see Section IV-C).

Laughter capability has been added to the Living Actor character production tools and rendering component: specific facial morphing data are exported from 3D character animation tools and later rendered in real time. Laughter audio can be played from an audio file, which can either be the recording of a human laughter or a synthetic laughter synchronized with the real laughs. A user interface has been added to test various avatar behaviors and play sounds.

An Application Programming Interface has been added to the Laugh Machine Living Actor module to remotely control the avatar using BML scripts. A separate component was created in Java to make the interface between the Laugh Machine messaging system using ActiveMQ and TCP/IP messages of Living Actor API. At this stage, the supported BML instructions are restricted to a few commands, triggering

[7]The source code for these models is publicly available at http://www.matthewzeiler.com/software/RetargetToolbox/Documentation/index.html

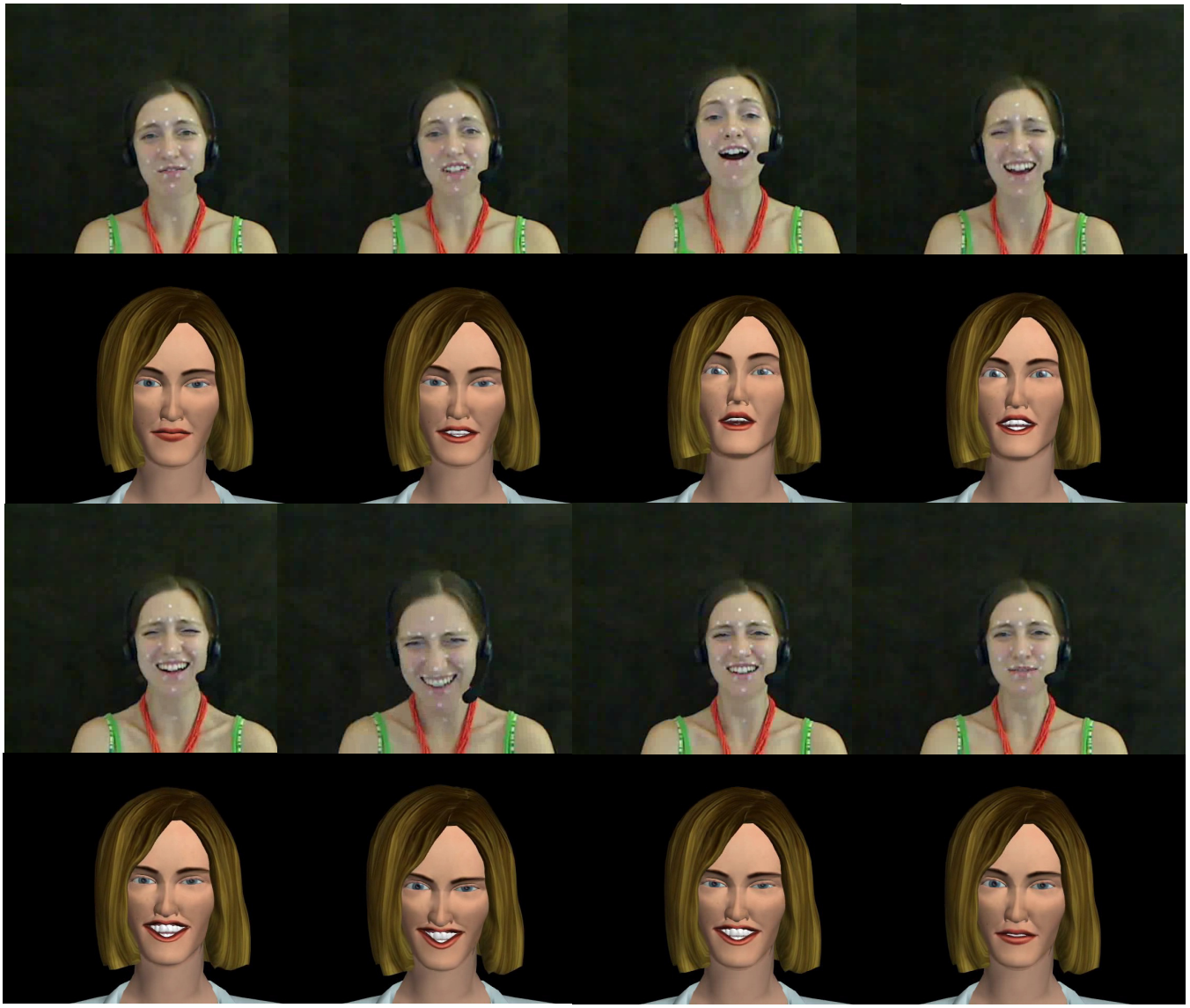


Fig. 13. Animation from Automatic Facial Movements detection

predefined laughs. But the foundation of more complex scripts is ready.

When there are no instructions sent, the real-time 3D rendering component automatically triggers "Idle" animations during which the virtual agent is breathing, making it more realistic and assuring animations continuity.

C. Audiovisual laughter synthesis

In the present work, no new laughter is generated. Instead, existing laughs are re-synthesized. All the animations can thus be prepared. For all the laughter animations, we synthesized separately the acoustic and the visual modalities, using the original audiovisual signals (with synchronized audio and video flows). In consequence the synthesized audio and video modalities are also synchronized. Each acoustic laugh was synthesized and the produced WAVE file was made available to the virtual agent. When the agent receives the instruction to laugh, it loads simultaneously the acoustic (WAVE) file and the BML animation file, and plays them synchronously.

IX. Experiments

A. Participants

Twenty-one participants (13 males; ages ranging from 25 to 56 years, $M = 33.16$, $SD = 8.11$) volunteered to participate. Four participants were assigned to the fixed speech condition, 5 to the fixed laughter condition and 11 to the interactive condition.

B. State and Trait influences on the perception of the virtual agent and its evaluation

Three kinds of subjective ratings were utilized to assess a) habitual and b) actual factors affecting the perception of the virtual agent and c) the evaluation of the interaction. For the habitual factors, two concepts were used: the dispositions towards ridicule and laughter, and the temperamental basis of the sense of humor, with one questionnaire each (PhoPhiKat$< 45 >$; [42]; State-Trait Cheerfulness Inventory,

STCI; [43]). Actual factors were assessed by measuring participant's mood before and after the experiment (state version of the STCI; [44]). The evaluation of the interaction was assessed with the Avatar Interaction Evaluation Form (AIEF; [45]).

1) Habitual Factors:

The assessment of personality variables allowed for a control of habitual factors influencing the perception of the virtual agent, independent of its believability. For example, gelotophobes, individuals with a fear of being laughed at (see [46]), do not perceive any laughter as joyful or relaxing and they fear being laughed at even in ambiguous situations. Therefore, the laughing virtual agent might be interpreted as a threat and the evaluation would be biased by the individuals fear. By assessing the gelotophobic trait, individuals with at least a slight fear of being laughed at can either be excluded from further analysis, or the influence of gelotophobia can be investigated for the dependent variables. Further, the joy of being laughed at (gelotophilia) and the joy of laughing at others (katagelasticism) might alter the experience with the agent, as katagelsticists might enjoy laughing at the agent, while gelotophiles may feel laughed at by the agent and derive pleasures from this. Both dispositions may increase the positive experience of interacting with an agent. The PhoPhiKat-45 is a 45-item measure of gelotophobia ("When they laugh in my presence I get suspicious"), gelotophilia ("When I am with other people, I enjoy making jokes at my own expense to make the others laugh"), and katagelasticism ("I enjoy exposing others and I am happy when they get laughed at"). Answers are given on a 4-point Likert scale ($1 =$ strongly disagree to $4 =$ strongly agree). Ruch and Proyer [42] found high internal consistencies (all alphas $\geq .84$) and high retest-reliabilities $\geq .77$ and $\geq .73$ (three to six months). In the present sample, reliabilities were satisfactory to high and ranged between α = .81 to .83.

Also, it was shown that the traits and states representing the temperamental basis of the sense of humor influence an individual's threshold for smiling and laughter, being amused, appreciating humor or humorous interactions (for an overview see [47]). It was assumed that trait cheerful individuals would enjoy the interaction more than low trait cheerful individuals, as they have a lower threshold for smiling and laughter, those behaviors are more contagious and there are generally more elicitors of amusement to individuals with high scores. For trait bad mood, it was expected that individuals with high scores would experience less positive affect when interacting with the agent, compared to individuals with low scores, as individuals with high scores have an increased threshold for being exhilarated, and they do not easily engage in humorous interactions.

The STCI assesses the temperamental basis of the sense of humor in the three constructs of cheerfulness (CH), seriousness (SE), and bad mood (BM) as both states (STCI-S) and traits (STCI-T). Participants completed the STCI-T before the experiment to be able to investigate the influence of cheerfulness, seriousness and bad mood on the interaction. The standard state form (STCI-S$<30>$; [44]) assesses the respective states of cheerfulness, seriousness and bad mood with ten items each (also on a four-point answering scale). Ruch and Köhler [48] report high internal consistencies for the traits (CH: .93, SE: .88, and BM: .94). The one month test-retest stability was high for the traits (between .77 and .86), but low for the states (between .33 and .36), conforming the nature of enduring traits and transient states.

2) Actual Factors:

Different experiments and studies on the state-trait model of cheerfulness, seriousness, and bad mood showed that participant's mood alters the experience of experimental interventions and natural interactions (for an overview, see [47]). Also, individual's mood changes due to interactions and interventions, for example state seriousness and bad mood decrease when participating carnival celebrations, while cheerfulness increases. Therefore, state cheerfulness, seriousness and bad mood were assessed before and after the experiment to investigate mood influence on the interaction with the agent (with the above mentioned STCI-S).

3) Evaluation:

To evaluate the quality of the interaction with the virtual agent, the naturalness of the virtual agent and cognitions and beliefs toward it, a questionnaire was designed for the purposes of the experiment. The aim of the Avatar Interaction Evaluation Form (AIEF) is to assess the perception of the agent, the emotions experienced in the interaction, as well as opinions and cognitions towards it on broad dimensions. The instrument consists of 32 items and 3 open questions, which were developed following a rational construction approach. The first seven statements refer to general opinions/beliefs and feelings on virtual agents (*e.g.*, "generally I enjoy interacting with virtual agents"). Then, 25 statements are listed to evaluate the experimental session. The following components are included: positive emotional experience (8 items; *e.g.*, "the virtual agent increased my enjoyment"), social (and motivational) aspects (7 items; *e.g.*, "being with the virtual agent just felt like being with another person"), judgment of technical features of the virtual agent/believeability (5 items; *e.g.*, "the laughter of the virtual agent was very natural"), cognitive aspects assigned to the current virtual agent (5 items; *e.g.*, "the virtual agent seemed to have a personality"). All statements are judged on a seven point Likert-scale (1 = strongly disagree to 7 = strongly agree). In the three open questions, participants can express any other thoughts, feelings or opinions they would like to mention, as well as describing what they liked best/least.

4) Further Evaluation Questions and Consent Form:

To end the experimental session, the participants were asked for general questions to assess their liking of candid camera humor in general ("Do you like candid camera-clips in general?" "How funny were the clips?" "How aversive were the clips?" "Would you like to see more clips of this kind?"). All questions were answered on a seven point Likert-scale. Then, participants were asked to give written consent to the use of the collected data for research and demonstration purposes (eNTERFACE workshop and ILHAIRE[8] project).

C. Conditions

1) Overview:

[8]http://www.ilhaire.eu

To create an interaction setting, the participants were asked to watch a film together with the virtual agent. Three conditions were designed (fixed speech, fixed laughter, interactive), systematically altering the degree of expressed appreciation of the clip (amusement) in verbal and non-verbal behavior, as well as different degrees of interaction with the participant's behavior. In the fixed speech and fixed laughter conditions, the agent would be acting independent of the participant, but still be signaling appreciation. In the interactive condition, the agent was responding to the participant's behavior. In other words, only the contextual information was used in the fixed speech and fixed laughter conditions, while the input and decision components (see Sections VI and VII) were active in the interactive condition.

2) Selection of pre-defined time points for the fixed laughter and fixed speech condition:

The pre-defined times were chosen from the stimulus video. Firstly, 14 subjects (three females) watched the video material and annotated the funniness to it on a continuous funniness rating scale (ranging from "not funny at all" to "slightly funny", to "funny", to "really funny" to "irresistibly funny"). Averaged and normalized funniness scores were computed over all subjects, leading to sections with steep increases in funniness (apexes; see Figure 14) over the video. Secondly, the trained raters assigned "punch lines" to the stimulus material, basing on assumptions of incongruity-resolution humor theory. Whenever the incongruous situation/prank was resolved for the subject involved, and amusement in the observer would occur from observing the resolution moment, a peak punch line was assigned. Punch lines were assigned for the first punch line occurring and the last punch line occurring in a given clip. When matching the continuous ratings with the punch lines, it was shown that the funniness apexes did cluster within the first and last punch lines for all subjects and all pranks, apart from one outlier. Table II shows the overall and apex durations of each clip, as well as the number and intensity of the peaks that have been fixed. For the three long apex sections, two responses were fixed, were the averaged funniness ratings peaked. Those peaks were rated on an intensity scale from 1 to 4. Pre-defined time points were controlled for a 1.5s delay in the rating/recording, due to reaction latency of the subjects and motor response delay.

TABLE II
DURATION, APEX AND NUMBER FO FIXED RESPONSES FOR EACH OF THE STIMULUS CLIPS

Clip	Duration (s)	Apex duration (s)	Fixed responses	Intensity
1	95	69	2	4
2	131	56	2	2
3	72	26	1	4
4	72	16	1	3
5	78	50	2	1

Notes: 1. Duration of apex (1st to last punch line). 2. Intensity (1 = strong; 4 = weak)

3) Fixed Speech:

In the fixed speech condition, the agent expressed verbal appreciation in 8 short phrases (*e.g.*, "oh, that is funny", "I liked that one", "ups", "this is great", "how amusing", "phew", nodding, "I wonder what is next") at pre-defined times. The verbal responses were rated for intensity on a four point scale and matched to the intensity scores of the pre-defined time points.

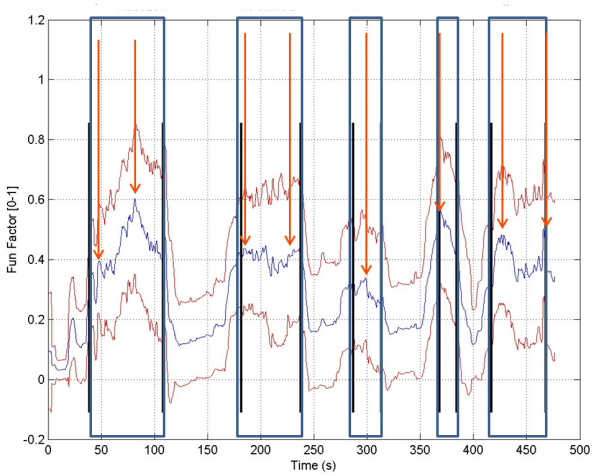


Fig. 14. Continuous funniness ratings (means in blue and standard deviations in red) over the stimulus video for 14 subjects and expert assigned punch lines (first and last, in blue) to each clip. Red arrows indicate time points for fixed responses.

4) Fixed Laughter:

In the fixed laughter condition, the agent laughed at pre-defined times during the video. The times were the same as the time points in the fixed speech condition. The agent displayed 8 laughs which varied in intensity and duration, according to the intensity ratings of the pre-defined time points. A laughter bout may be segmented into an onset (*i.e.*, the pre-vocal facial part), an apex (*i.e.*, the period where vocalization or forced exhalation occurs), and an offset (*i.e.*, a post-vocalization part; often a long-lasting smile fading out smoothly; see [21]). Therefore, the onset was emulated by an independent smiling action just before the laughter (apex) would occur at the fixed time. The offset of the laughter was already integrated in the 8 laughter chosen.

5) Interactive Condition:

In the interactive condition, which follows the architecture presented in Section V and Figure 1, the agent was using two sources of information to respond to the participant: the continuous funniness ratings to the clip (context, shown in Figure 14) and the participant's acoustic laughter vocalizations. The dialog manager was receiving these two information flows and continuously taking decisions about whether and how the virtual agent had to laugh, providing intensity and duration values of the laugh to display. These instructions were then transmitted to the audiovisual synthesis modules. Due to the limited number of laughs available for synthesis (14 at the time of the experiments), it was decided to cluster them into 4 groups based on their intensities and durations. The output of the dialog manager is then pointing to one of the clusters, inside which the laugh to synthesize is randomly picked.

D. Problems encountered

Several problems appeared during the experiments.

First of all, the computers we used were not powerful enough to run all the components on a single computer. We had to connect four computers together: one master computer running the stimulus video and the Kinect recording and analysis (+ the context), one computer running a webcam with shoulder movement tracking driven by Eyesweb, another one running the dialog manager and finally one computer for displaying the virtual agent. Still, the master computer could not record the video stream from the Kinect. We decided to run the experiments without recording that video as we still have the webcam recording, but this issue should be investigated in the future. Furthermore, during some experiments, data transmission from one computer to the other was suffering from important delays (5-10s), which obviously affect the quality of the interaction. Reducing these delays will be one of the most important future developments.

Second, the audio detection module had been trained with data containing mostly laughs, and relatively few other noises. Hence, there was confusion between laughter and other loud noises. In addition, the detection was audio-only, which does not enable to take smiles or very subtle laughs (with low audio) into account. We are already working on improving the laughter detection and including other modalities (video, respiration) to increase its robustness.

Third, from the training data, it appeared that the context was by far the best factor to explain participants' laughs: in consequence, the dialog manager did not pay attention to what the participant was doing, but only triggered laughs from the contextual input. Since this is undesirable behavior in the interactive condition (which is in that case actually similar to the fixed laughter condition, as every reaction is only context-dependent), we decided to omit the context in the interactive condition: the virtual agent was then only reacting to what the participant was doing. Better models should be built in the future to allow both context and participant's reactions to be considered simultaneously.

Fourth, the pool of available laughs for synthesis is currently limited. There are not a lot of laughs from one single voice for which we have good quality synthesis for both the audio and the visual modalities. This limits the range of actions the virtual agent is able to perform and some participants with whom the agent laughed a lot might have noticed some repetitions. This will be improved in the future with 2 solutions: 1) a larger pool of available laughs 2) the possibility to generate new laughter transcription and/or modify existing ones in real-time.

Finally, a connection problem with the respiration sensor prevented us from recording respiration data.

E. Procedure of the evaluation study

Participants were recruited through e-mail announcement of an "evaluation study of the Laugh Machine project" at the eNTERFACE workshop. As an incentive, participants were offered a feedback on the questionnaire measures on request. It was announced that the study consisted of the filling in of questionnaires (approximately 30-45 minutes) and a session of 30 minutes on two given days. No further information on the aims of the study was given. Participants chose a date for the experimental session via the Internet and received confirmation by email.

At the experimental session, participants were welcomed by one of the two female experimenters and asked to fill in the STCI and the PhoPhiKat. Then, participants were asked to fill in the STCI-S to assess their current mood. Meanwhile, the participants were assigned to one of the three conditions. Afterwards, the second female experimenter accompanied the participant to the experimenting room, where the participant was asked to sit in front of a television screen. A camera allowed for the frontal filming of the head and shoulder, as well as upper body of the participant. Two male experimenters concerned with the technical components were present. Participants were asked for consent to have their shoulder and body movements recorded. They were also given headphones to hear the virtual agent. The experimenter explained that the participant was asked to watch a film together with Poppy and that the experimenters would leave the room when the experiment started. Once the experimenters left the room, the agent did greet the participant ("Hi, I'm Greta. I'm looking forward to watch this video with you. Let's start") and subsequently, the video started. After the film, the experimenters entered the room again and the female experimenter accompanied the participant back to the location where the post measure of the STCI-S, as well as the AIEF and five further evaluation questions were filled in. After all questionnaires were completed, the first female experimenter debriefed the participant and asked for written permission to use the obtained data.

The following setup was used in this experimental session (see Figure 15). Two LCD displays were used: the bigger one (46") was used to display the stimuli (the funny film, see Section III). The smaller (19") LCD display placed on the right side of the big one was used to display the agent (a close-up view of the agent with only the face visible was used). Four computers were used to collect the user data, run the Dialog Module and to control the agent audio-visual synthesis. Participant's behaviors were collected through a Kinect (sound, depth map, and camera) and a second webcam synchronized with the EyesWeb software (see Section VI). Because of technical issues we were not able to use the respiration sensor in this experimental session. Participants were asked to sit on a cushion about 1m from the screen. They were asked to wear headphones.

In the evaluation we have used 14 laugh episodes from the AVLC dataset (subject 5). For consistency reasons we have used only one female agent (*i.e.*, Poppy) and the animation created with only one method *i.e.* automatic facial movements detection (see Section VIII-B3).

X. Results

A. Preliminary Analysis

Scale means for cheerfulness, seriousness, bad mood, gelotophobia, gelotophilia and katagelasticism were investigated. The sample characteristics of the PhoPhiKat and the STCI-T

Fig. 15. Setup of the experiment.

resembled norm scores for adult populations. In this sample, the internal consistencies were satisfactory for all trait scales, ranging from $\alpha = .74$ for trait seriousness, to $\alpha = .91$ for trait cheerfulness. In respect to trait variables biasing the evaluation, three subjects were identified for exceeding the cut-off point for gelotophobia. Means for the state cheerfulness, seriousness and bad mood scores showed higher state bad mood scores before the experiment, compared to previous participants of studies on personality and humor. In respect to the AIEF, the internal consistencies (Cronbach's alpha) of the scales were satisfactory, ranging from $\alpha = .78$ (cognitive aspects) to $\alpha = .90$ (positive emotional experience).

B. Traits

In line with previous findings, trait cheerfulness was correlated negatively to trait bad mood ($r = -.61, p < .01$), as well as trait seriousness ($r = -.16, n.s.$), but less strongly to the latter one. Trait seriousness and bad mood were correlated positively ($r = .22, n.s.$). Gelotophobia was correlated negatively to gelotophilia ($r = -.50, p < .05$), as well as (but less so) to katagelasticism. The latter negative (but not significant; $r = -.35, p = .117$) correlation was unusual, as gelotophobia usually shows zero correlations to katagelasticism. Katagelasticism was positively related to gelotophilia ($r = .26, n.s.$). Generally, correlations of the AIEF to the trait scale did not reach statistical significance. Correlating the dimensions and items of the AIEF to gelotophobia (bivariate Pearson correlations) showed, that three of the four AIEF scales were negatively correlated with gelotophobia, indicating that higher scores in gelotophobia went along with less positive emotions, less assignment of cognition and less believability of the virtual agent to participants with higher scores. Feeling social presence by the agent was positively correlated to gelotophobia. Gelotophilia correlated positively with all dimensions of the AIEF. Further, higher scores in katagelasticism went along with more positive emotions, higher perceived believability and higher perceived social presence. With regard to the temperamental basis of the sense of humor, the highest correlations to the AIEF dimensions were found for trait bad mood. Unlike a priori assumptions, trait bad mood correlated positively to the AIEF dimensions and correlations to trait cheerfulness were generally very low. Trait seriousness was correlated negatively to the AIEF scales.

C. States

Correlating the states to their respective traits showed that trait cheerfulness was positively correlated to state cheerfulness both pre and post the experiment (but all $n.s.$). Trait seriousness was positively correlated to seriousness after the experiment, whereas trait bad mood was negatively correlated to both, bad mood pre and post the experiment (both $p < .01$). In this sample, a few individuals with low scores in trait bad mood came to the experiment with high values in state bad mood, whereas a few individuals with high scores on bad mood came to the experiment with comparably low scores in state bad mood. Descriptive analysis of the mood before and after the experiment, it was found that the interaction with the virtual agent led to a decrease in seriousness over all conditions, whereas state cheerfulness stayed stable. State bad mood before the experiment predicted lower scores on the AIEF dimensions, suggesting that individuals that feel more grumpy or sad generally experience less positive emotions with the virtual agent, assign the agent less cognitive capability, experience less social interaction and judge it as less believable (all $r < -.489$, $p < .05$).

D. AIEF Scales/Dimensions

Due to the low cell sizes, no test of significance could be performed to testing the influence of the condition on the AIEF dimensions. Nevertheless descriptive inspection of the group means showed that the conditions differed in their elicitation of positive outcomes on all dimensions of the AIEF. The interactive condition yielded highest means on all four dimensions, implying that the participants felt more positive emotions, felt more social interaction with the agent, considered it more natural and assigned it more cognitive capabilities than in the fixed conditions (see Figure 16). The means of the interactive condition were followed by the means of the fixed laughter condition.

Interestingly, the fixed speech condition yielded similarly high scores on the beliefs on cognition as the interactive condition, whereas the other means were numerically lowest for the fixed speech condition.

The results stayed stable when excluding the three individuals exceeding the cut-off point for gelotophobia.

E. Open Answers

Out of the 21 participants, 14 gave answers to the question of what they liked least about the session. Half of the participants mentioned that the video was not very funny or would have been funnier with sound. Two participants mentioned that they could not concentrate on both, the virtual agent and the film. Two of the three gelotophobes gave feedback (subject 2: "Poppy's expression while laughing was more a smirk than a

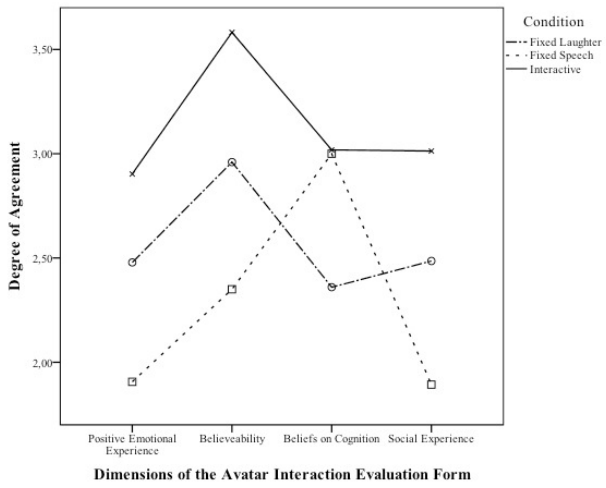


Fig. 16. Profiles of the means in the AIEF scales for the three experimental conditions separately

laugh"; subject 21: "it's hard to act naturally when watching a film when you feel like you should laugh"). Seventeen participants responded to what was liked best about the session. Best liked was the laughter of the virtual agent through the headphones (it was considered amusing and contagious; three nominations), the video (five nominations), the set up (four nominations) and one participant stated: "It was interesting to see in what situations and in what manner the virtual agent responded to my laughter and to funny situations respectively" (subject 12).

XI. Collected Data

The multimodal laughter corpora of human to human interactions are rare. Even more seldom are corpora of human-machine interaction that contain any episodes of laughter. The evaluation of our interactive laughter system gave us the unique opportunity to gather new data about the human behavior in such human-machine interactive scenario. Consequently, we have collected multimodal data from participants to our experiments. In more details our corpus contains:

- audio data, recorded by the Kinect at 16kHz and stored in mono WAVE files, PCM 16bits
- Kinect depth maps
- two web cameras
- data on the shoulders movement extrapolated from the video stream (for this purpose two small markers were placed on the shoulders of each participant)

All these data can be synchronized with the context (see Section IX-C2) and the agent reactions. The collected corpus is an important result of the Laugh Machine Project. It will be widely used in the ILHAIRE project and will become freely available for the research purposes.

XII. Conclusions

The first results of the evaluation experiment are highly promising: it was shown that the three conditions elicited different degrees of positive emotions in the participants, the amount of social interaction induced, as well as the cognitions and capability assigned to the agent. Also, the believability differed for all three conditions. It was shown that the interactive condition yielded the most positive outcomes on all dimensions, implying that the feedback given to the participant by mimicking his or her laughter is best capable of creating a "mutual film watching experience" that is pleasurable.

In sum, expressing laughter increases the positive experience in the interaction with an agent, when watching an amusing video (both laughter conditions elicited more positive emotions), compared to the fixed speech condition. The fixed speech condition yielded numerically lowest means on the AIEF dimensions, apart from the dimension "beliefs on cognition", where the means where as high as in the interactive condition, implying that speech leads to the assignment of cognitive ability equally as much as responding to the participant's behavior. Naturally, the fixed speech conditions should yield the lowest scores, as there was no laughter expressed in this conditions and some items targeted the contagiousness and appropriateness of the laughter displayed by the agent.

Obviously, in the interactive condition, the amount of laughter displayed by the agent varied for each participant, depending on how many times the participants actually laughed. Therefore, the agent behaved similar to the participant, which seems to be natural and comfortable for the participant. Nevertheless, the current state of data analysis does not allow to differentiating between individuals who displayed a lot of laughter—and consequently had a lot of laughter feedback by the agent—and individuals who showed only little laughter—and received little laughter feedback by the agent. An in depth analysis of the video material obtained during the eNTERFACE evaluation experiment will allow for an investigation of how many times the participants actually laughed and how this influenced the perception of the setting. This will be done by applying the FACS [39]. Further, an analysis of the eye movements (gaze behavior) will allow for an estimation of the attention paid to the agent.

The results of the trait and state cheerfulness, seriousness, and bad mood variables clearly show the importance of including personality variables into such evaluation experiments. Especially state bad mood influenced the interaction and latter perception of the virtual agent, leading to a mood dependent bias. Individuals with high scores in state bad mood before the experiment evaluated the virtual agent less favorably. This is likely due to their enhanced threshold for engaging in cheerful/humorous situations/interactions and—in the case of grumpiness—their unwillingness to be exhilarated and—in the case of depressed/melancholic mood—the incapability to be exhilarated. Therefore, personality should always be controlled for in future studies. Generally, there was sufficient variance in the gelotophobia scores, even in the little sample obtained in the evaluation. Gelotophobia showed some systematic relations to the dimensions of the AIEF. For future studies, the assessment and control of gelotophobia is essential to get unbiased evaluations of an agent. Furthermore, those results might help the understanding of the fear of being laughed at and how it influences the thoughts, feelings and behavior of

individuals with gelotophobia.

Nevertheless, more participants are needed to test the hypothesis on the influence of the condition on the AIEF dimensions in order for any statistically significant differences between the conditions to be found. To improve the experimental set up, challenges from eNTERFACE, as well as the participant's feedback will be integrated to optimize the design and procedure. For example, the stimulus video consisted of only one type of humorous material. It is well established in psychological research that inter-individual differences exist in the appreciation of types of humor. Therefore, a lack of experienced amusement on the side of the participant might also be due to the disliking of candid camera clips, as one specific type of humor. Any manipulation by the experimental conditions should not be overshadowed by the quality/type of stimulus video. Therefore, a more representative set of clips with sound is needed (presented in counter-balanced order, also extending the overall interaction time with the virtual agent).

Furthermore, it needs to be clear to participants beforehand, what the virtual agent is capable of doing. In the beginning of the experiment, the virtual agent should display some laughter, so the participant knows, that the virtual agent would be capable of showing this behavior. This ensures, that the participant will not be solely surprised and amused by the fact, the virtual agent can laugh, when it eventually does during the course of the film. If this information is not available to participants, it might be that the amusement is only due to the excitement/pleasure of the technical development of making a virtual agent laugh. Ruch and Ekman's [21] overview on the knowledge about laughter (respiration, vocalization, facial action, body movement) illustrated the mechanisms of laughter, and defined its elements. While acknowledging that more variants of this vocal expressive-communicative signal might exist, they focused on the common denominators of all forms but proposed distinguishing between laughing spontaneously (emotional laughter) and laughing voluntarily (contrived or faked laughter). In this experiment, only displays of amusement laughter (differing in intensity and duration) were utilized. Further studies may also include different variants of types of laughter.

On the technical side, the biggest outcome of the project is a full processing chain with components that can communicate together to perform multimodal data acquisition, real-time laughter-related analysis, output laughter decision and audio-visual laughter synthesis. Progresses have been accomplished on all these aspects during the Laugh Machine project. We can cite the development of the respiration sensor and the integration of all input devices in a synchronized framework, which will enable multimodal laughter detection; the construction of a real-time, speaker independent, laughter intensity estimator; the design of the first dialog manager dealing with laughter; the first advances in acoustic laughter synthesis with the introduction of HMM-based processes; the four different animation techniques that have been implemented; or the unique database of humans interacting with a laughing virtual agent that has been collected.

Each of these components can be improved and several issues arose during the experiments. Without going into details for each of the involved components, future works will include: improving the laughter detection and intensity computation with the help of visual and respiration signals; reducing the communication delays between the computers hosting the different modules; better balancing the influence of the context in the dialog manager; extending the range of output laughs by allowing laughs to be generated or modified on the fly; ensuring that all experimental data can be recorded flawlessly; or adapting the virtual agent's behavior to the participant's personality (*e.g.*, gelotophobe) and mood to maximize the participant's perception of the interaction. Also, future agents may not only include facial expressions and vocal utterances, as laughter also entails lacrimation, respiration, body movements (*e.g.*, [49]), body posture and vocalization.

However, despite all the identified issues, the first evaluation results are positive. This is very encouraging and indicates that the full LaughMachine system, while imperfect, is already working and providing us with both a nice benchmark and a reusable framework to evaluate future developments.

ACKNOWLEDGMENT

This work was supported by the European FP7-ICT projects ILHAIRE (FET, grant $n^\circ 270780$), CEEDS (FET, grant $n^\circ 258749$) and TARDIS (STREP, grant $n^\circ 288578$). The authors would also like to thank the organizers of the eNTERFACE'12 Workshop for making the project possible and making available high quality material and recording rooms to us. Finally, all the participants of our experiments are gratefully acknowledged.

REFERENCES

[1] L. Kennedy and D. Ellis, "Laughter detection in meetings," in *NIST ICASSP 2004 Meeting Recognition Workshop*, Montreal, May 2004, pp. 118–121.

[2] K. P. Truong and D. A. van Leeuwen, "Automatic discrimination between laughter and speech," *Speech Communication*, vol. 49, pp. 144–158, 2007.

[3] M. T. Knox and N. Mirghafori, "Automatic laughter detection using neural networks," in *Proceedings of Interspeech 2007*, Antwerp, Belgium, August 2007, pp. 2973–2976.

[4] S. Petridis and M. Pantic, "Fusion of audio and visual cues for laughter detection," in *Proceedings of the 2008 international conference on Content-based image and video retrieval*. ACM, 2008, pp. 329–338.

[5] ——, "Audiovisual discrimination between laughter and speech," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Las Vegas, Nevada, 2008, pp. 5117–5120.

[6] ——, "Audiovisual laughter detection based on temporal features," in *Proceedings of the 10th international conference on Multimodal interfaces*. ACM, 2008, pp. 37–44.

[7] S. Petridis, A. Asghar, and M. Pantic, "Classifying laughter and speech using audio-visual feature prediction," in *Proceedings of the 2010 IEEE International Conference on Acoustics Speech and Signal Processing (ICASSP)*. Dallas, Texas: IEEE, 2010, pp. 5254–5257.

[8] S. Sundaram and S. Narayanan, "Automatic acoustic synthesis of human-like laughter," in *Journal of the Acoustical Society of America*, vol. 121, no. 1, January 2007, pp. 527–535.

[9] E. Lasarcyk and J. Trouvain, "Imitating conversational laughter with an articulatory speech synthesis," in *Proceedings of the Interdisciplinary Workshop on The Phonetics of Laughter*, August 2007, pp. 43–48.

[10] T. Cox, "Laughter's secrets: faking it – the results," *New Scientist*, 27 July 2010. [Online]. Available: http://www.newscientist.com/article/dn19227-laughters-secrets-faking-it--the-results.html

[11] P. DiLorenzo, V. Zordan, and B. Sanders, "Laughing out loud: control for modeling anatomically inspired laughter using audio," in *ACM Transactions on Graphics (TOG)*, vol. 27, no. 5. ACM, 2008, p. 125.

[12] D. Cosker and J. Edge, "Laughing, crying, sneezing and yawning: Automatic voice driven animation of non-speech articulations," in *Proc. of Computer Animation and Social Agents (CASA09)*. Citeseer, 2009, pp. 21–24.

[13] J. Urbain, R. Niewiadomski, E. Bevacqua, T. Dutoit, A. Moinet, C. Pelachaud, B. Picart, J. Tilmanne, and J. Wagner, "AVLaughterCycle: Enabling a virtual agent to join in laughing with a conversational partner using a similarity-driven audiovisual laughter animation," *Journal on Multimodal User Interfaces*, vol. 4, no. 1, pp. 47–58, 2010.

[14] S. Shahid, E. Krahmer, M. Swerts, W. Melder, and M. Neerincx, "Exploring social and temporal dimensions of emotion induction using an adaptive affective mirror," in *27th international conference extended abstracts on Human factors in computing systems*. ACM, 2009, pp. 3727–3732.

[15] S. Fukushima, Y. Hashimoto, T. Nozawa, and H. Kajimoto, "Laugh enhancer using laugh track synchronized with the user's laugh motion," in *Proceedings of the 28th of the international conference on Human factors in computing systems (CHI'10)*, 2010, pp. 3613–3618.

[16] C. Becker-Asano, T. Kanda, C. Ishi, and H. Ishiguro, "How about laughter? Perceived naturalness of two laughing humanoid robots," in *Affective Computing and Intelligent Interaction*, 2009, pp. 49–54.

[17] C. Becker-Asano and H. Ishiguro, "Laughter in social robotics - no laughing matter," in *Intl. Workshop on Social Intelligence Design (SID2009)*, 2009, pp. 287–300.

[18] G. Mckeown, M. F. Valstar, R. Cowie, and M. Pantic, "The semaine comary:2007kkrpus of emotionally coloured character interactions," in *Proceedings of IEEE Int'l Conf. Multimedia, Expo (ICME'10), Singapore*, July 2010, pp. 1079–1084.

[19] J. Urbain, E. Bevacqua, T. Dutoit, A. Moinet, R. Niewiadomski, C. Pelachaud, B. Picart, J. Tilmanne, and J. Wagner, "The AVLaughterCycle database," in *Proceedings of the Seventh conference on International Language Resources and Evaluation (LREC'10)*, Valletta, Malta, May 2010.

[20] J. Urbain and T. Dutoit, "A phonetic analysis of natural laughter, for use in automatic laughter processing systems," in *International Conference on Affective Computing and Intelligent Interaction (ACII2011)*, Memphis, Tennesse, October 2011, pp. 397–406.

[21] W. Ruch and P. Ekman, "The expressive pattern of laughter," in *Emotion, qualia and consciousness*, A. Kaszniak, Ed. Tokyo: World Scientific Publishers, 2001, pp. 426–443.

[22] The Apache Software Foundation, "Apache ActiveMQ™ [computer program webpage]," http://activemq.apache.org/, consulted on August 24, 2012.

[23] J. Wagner, F. Lingenfelser, and E. André, "The social signal interpretation framework (SSI) for real time signal processing and recognition," in *Proceedings of Interspeech 2011*, 2011.

[24] N. Chawla, K. Bowyer, L. Hall, and W. Kegelmeyer, "Smote: synthetic minority over-sampling technique," *Journal of Artificial Intelligence Research*, vol. 16, pp. 321–357, 2002.

[25] F. Eyben, M. Wöllmer, and B. Schuller, "Opensmile: the munich versatile and fast open-source audio feature extractor," in *Proceedings of the international conference on Multimedia*, ser. MM '10. New York, NY, USA: ACM, 2010, pp. 1459–1462.

[26] R. Niewiadomski, J. Urbain, C. Pelachaud, and T. Dutoit, "Finding out the audio and visual features that influence the perception of laughter intensity and differ in inhalation and exhalation phases," in *Proceedings of the ES 2012 4th International Workshop on Corpora for Research on EMOTION SENTIMENT & SOCIAL SIGNALS, Satellite of LREC 2012*, Istanbul, Turkey, May 2012.

[27] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. Witten, "The weka data mining software: an update," *ACM SIGKDD Explorations Newsletter*, vol. 11, no. 1, pp. 10–18, 2009.

[28] M. Filippelli, R. Pellegrino, I. Iandelli, G. Misuri, J. Rodarte, R. Duranti, V. Brusasco, and G. Scano, "Respiratory dynamics during laughter," *Journal of Applied Physiology*, vol. 90, no. 4, p. 1441, 2001.

[29] A. Feleky, "The influence of the emotions on respiration." *Journal of Experimental Psychology*, vol. 1, no. 3, pp. 218–241, 1916.

[30] A. Camurri, P. Coletta, G. Varni, and S. Ghisio, "Developing multimodal interactive systems with eyesweb xmi," in *Proceedings of the 2007 Conference on New Interfaces for Musical Expression (NIME07)*, 2007, p. 302305.

[31] B. Lukas and T. Kanade, "An iterative image registration technique with an application to stereo vision," in *Proceedings of the 7th international joint conference on Artificial intelligence*, 1981.

[32] D. Winter, "Biomechanics and motor control of human movement," 1990.

[33] W. Sethares and T. Staley, "Periodicity transforms," *Signal Processing, IEEE Transactions*, vol. 47, no. 11, pp. 2953–2964, 1999.

[34] K. Tokuda, H. Zen, and A. Black, "An HMM-based speech synthesis system applied to English," in *IEEE Speech Synthesis Workshop*, Santa Monica, California, September 2002, pp. 227–230.

[35] K. Oura, "HMM-based speech synthesis system (HTS)," http://hts.sp.nitech.ac.jp/, consulted on June 22, 2011.

[36] T. Yoshimura, "Simultaneous modeling of phonetic and prosodic parameters, and characteristic conversion for hmm-based text-to-speech systems," Ph.D. dissertation, Ph. D. thesis, Nagoya Institute of Technology, 2002.

[37] T. Drugman and T. Dutoit, "The deterministic plus stochastic model of the residual signal and its applications," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 20, pp. 968–981, 2012.

[38] H. Kawahara, "Straight, exploitation of the other aspect of vocoder: Perceptually isomorphic decomposition of speech sounds," *Acoustical science and technology*, vol. 27, no. 6, pp. 349–353, 2006.

[39] P. Ekman, W. Friesen, and J. Hager, "Facial action coding system: A technique for the measurement of facial movement," 2002.

[40] J. Saragih, S. Lucey, and J. Cohn, "Deformable model fitting by regularized landmark mean-shift," *International Journal of Computer Vision*, vol. 91, no. 2, pp. 200–215, 2011.

[41] M. Zeiler, G. Taylor, L. Sigal, I. Matthews, and R. Fergus, "Facial expression transfer with input-output temporal restricted boltzmann machines," in *Neural Information Processing Systems Conference NIPS 2011*, 2011, pp. 1629–1637.

[42] W. Ruch and R. Proyer, "Extending the study of gelotophobia: On gelotophiles and katagelasticists," *Humor: International Journal of Humor Research*, vol. 22, no. 1-2, pp. 183–212, 2009.

[43] W. Ruch, G. Köhler, and C. Van Thriel, "Assessing the "humorous temperament": Construction of the facet and standard trait forms of the state-trait-cherrfulness-inventory–STCI." *Humor: International Journal of Humor Research; Humor: International Journal of Humor Research*, vol. 9, pp. 303–339, 1996.

[44] W. Ruch, G. Köhler, and C. Van Thriel, "To be in good or bad humour: Construction of the state form of the state-trait-cheerfulness-inventory–STCI," *Personality and Individual Differences*, vol. 22, no. 4, pp. 477–491, 1997.

[45] J. Hofmann, T. Platt, and W. Ruch, "Avatar interaction evaluation form (AIEF)," 2012, unpublished research instrument.

[46] W. Ruch and R. Proyer, "The fear of being laughed at: Individual and group differences in gelotophobia." *Humor: International Journal of Humor Research*, vol. 21, pp. 47–67, 2008.

[47] W. Ruch and J. Hofmann, "A temperament approach to humor," in *Humor and health promotion*, P. Gremigni, Ed. New York: Nova Science Publishers, 2012.

[48] W. Ruch and G. Köhler, "A temperament approach to humor," in *The sense of humor: Explorations of a personality characteristic*, W. Ruch, Ed. Berlin: Mouton de Gruyter, 2007, pp. 203–230.

[49] G. Hall and A. Allińn, "The psychology of tickling, laughing, and the comic," *The American Journal of Psychology*, vol. 9, no. 1, pp. 1–41, 1897.

Human motion recognition based on videos

Dominique De Beul

***Abstract*—Human motion recognition is implemented in many real time applications. In this paper, a non real time human motion recognition was performed based on recording files (bvh files and Kinect) by using support vetor machines and artificial neural networks. The results showed that the artificial neural networks had a better recognition rate than the support vector machines.**

***Index Terms*—Artificial neural networks (ANN), human motion, Kinect, machine learning, non intrusive method, support vector machines (SVM).**

I. Introduction

The recognition of human motion takes more and more places in our life thanks to the technology evolution. Human motion recognition has applications in many domains such as robotics, visual surveillance, content-based video database query and retrieval, human-computer interaction. The human motion recognition identifies the actions performed by body movement of human beings [1].

Motion can be analysed by intrusive methods or by non intrusive methods. Intrusive methods are used by placing optical or magnetic items on the subject. These methods can disturb or influence the subject in action. Non intrusive methods which use 2D or 3D cameras eliminate this drawback but provide less accuracy.

The skeletal joints coordinates supplied by the Kinect 3D camera makes it easier to model the human body in movement. A lot of attention is being given to recognition in real time [2]–[6], but what about the recognition from files recorded. To our knowledge, not enough research has showed a deep interest in this direction. Our perspective is to create a data set representing motion from dancers. Questioning of all data will be undertaken by a graphical expression e.g drawing a motion. This paper is the first step to reach our goal.

Many methods are employed in human motion recognition. Let us mention the ontologies which represent the human motion by using concepts and semantic rules and by following dance notation such as Benesh [7] or Laban [8]. In machine learning, methods as artificial neural networks [9], support vector machines (SVM) [10], hidden markov models (HMMs) [11], K-nearest neighbor (KNN) [12] and Bayes classification [13] are commonly implemented.

In this context, the main goal of this work was to study human motion recognition coming from the Kinect and transformed in BVH files (Fig. 1). Another objective was to compare 2 recognition methods: artificial neural networks and support vector machines, in real time and in non real time.

D. De Beul is with the Computer Science Department, Faculty of Engineering, University of Mons, Belgium.

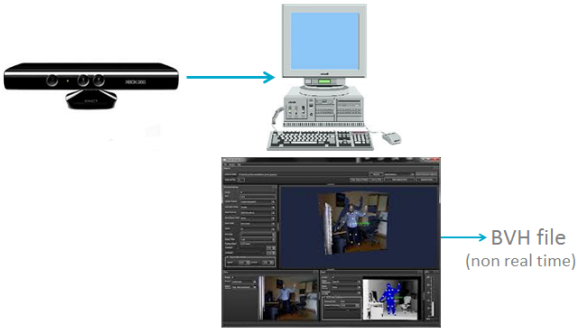


Fig. 1. Data transformation from Kinect to BVH files.

II. Materials and Methods

A. Data acquisition and processing

Data of human skeleton motion was supplied by the Kinect camera at the rate of 30 fps to the Brekel software (www.brekel.com) which changed them in BVH (Biovision hierarchy) files. To represent the skeleton, 23 joints per frame were used, and four types of motion were studied: the walking, the arms, the legs and the arms/legs motion (Fig. 2). Noted that only the walking motion moves the whole skeleton.

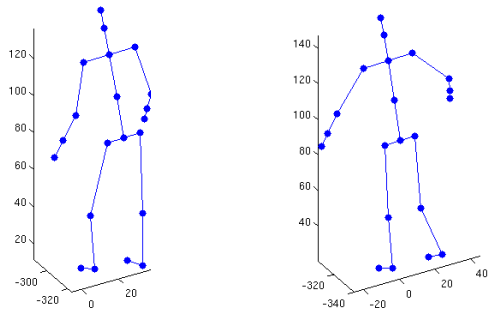


Fig. 2. Skeleton representations: examples of walking motion (left) and arms motion (right).

The BVH format was divided in 2 parts (fig. 3,4), the first specified the skeleton hierarchy such as the parent joints, the offset, and the second part provided motion data for each joint such as the Euler angles. Matlab R2011a (Mathworks) was employed to calculate the products of the transformation matrices of every parent joint (except for the root) to find the joint coordinates in the Kinect reference system [14]. The speed of every joint was taken into consideration for the temporality representation.

Seven people were involved in the video recordings, each had made 4 types of motion during 2 minutes. The 10 first seconds of 5 video recordings were taken for the learning

phase. For the test phase, 2 seconds were chosen randomly from the 2 others video recordings.

```
HIERARCHY
ROOT Hips
{
  OFFSET -7.166 21.603 -287.920
  CHANNELS 6 Xposition Yposition Zposition Zrotation
  Xrotation Yrotation
  JOINT LeftUpLeg
  {
    OFFSET 9.559 0.000 -0.000
    CHANNELS 3 Zrotation Xrotation Yrotation
    JOINT LeftLeg
    {
    ...
```

Fig. 3. Hierarchy part: in this example LeftUpLeg joint depends on the Hips and LeftLeg joint depends on the LeftUpLeg. Offsets are given and don't change during the motion.

```
MOTION
Frames: 3692
Frame Time:     0.0333333
-4.85759 92.5133 -283.24 -1.61071 17.1168 16.0006
-0.482831 6.58091 0.0553383 3.79237 -11.7427
0.674913 1.00486e-07 -1.81736e-08 4.7216e-07
1.00059 5.29579 -0.0923595 1.25669 -2.56676
0.100948 -7.13941e-08 -1.19396e-06 -4.64743e-07
-5.48669e-07 -1.77784e-06 1.47369e-05 0 0 0
16.2893 -3.20818 14.2277 -9.87954 -17.5379 3.7434
5.75159e-09 1.86537e-07 1.45938e-06 0 0 0 -16.9417
4.13677 11.1494 10.2383 -19.3523 10.3552
-6.76001e-08 3.94474e-09 -3.94002e-09 -0.212399
-1.11817 -0.00414335 1.15099e-07 -5.15878e-07
-9.242e-07
...
```

Fig. 4. Motion part: in this example the first frame is showed. After the frame quantity and the frame time (in seconds), 57 numbers represent the skeleton.

B. Supervised learning model: Support vector machines

Support Vector Machines (SVM) were used to discriminate between 2 binary classes linearly separable by maximizing the distance separating the hyperplane and the closest data so called hard margin [15]. The soft margin was obtained by relaxing constraints and by tolerating a margin error. Equation 1 represent the optimisation problem.

$$\begin{cases} \underset{w,b}{\text{minimize}} & \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{m}\xi_i \\ \text{subject to:} & y_i\left(\langle w \cdot x_i\rangle + b\right) \geq 1 - \xi_i \\ & \xi_i \geq 0 \\ & i = 1, \ldots, m \end{cases} \quad (1)$$

Where w represent the weight vectors, C the regularisation constant, ξ_i the margin error, (x_i, y_i) are the training set and $\langle w \cdot x_i\rangle$ is the inner product.

The model was calculated with the LIBSVM library [16] by using the soft margin with the radial basis function as kernel and n-fold cross validation. The strategy one against one was adopted for the multi-class learning.

C. Supervised learning model: Artificial Neural Networks

Artificial neural networks were employed for data classification. To learn, a network decreased the mean squared error (MSE) by using a backpropagation algorithm which adjusted the synaptic weights from each neurons [17]. Our networks were builded from the Matlab neuronal toolbox network.

The first network was created with 1 hidden layer of 10 neurons and the second with 2 hidden layers of 10 neurons for each layer. Data of 5 video recordings were employed, 70% for learning, 15% for validation and 15% for testing. Sigmoïdal functions and scaled conjugate gradient backpropagation algorithms were used for both networks.

III. Results

Ten-fold cross-validation were employed for the SVM model. A MSE=0.0044 at 318 epochs was determined in the validation phase for the neural network with 1 hidden layer and a MSE=0.0028 at 258 epochs (iterations) for the second neural network. Each type of motion was represented by 4 classes (CL1: Walking, CL2 : Arms, CL3 : Legs, C : Arms/Legs motion).

CL1	48(80%)	4	6	2	
CL2	0	57(96%)	0	3	
CL3	0	0	60(100%)	0	
CL4	0	1	3	56(94%)	
					221(92%)
	CL1	CL2	CL3	CL4	Total

Fig. 5. Confusion matrix (Test): SVM - C=512 and $\gamma = 0.03125$

CL1	221(100%)	0	0	0	
CL2	0	226(100%)	0	0	
CL3	3	0	238(98.8%)	0	
CL4	1	0	0	221(99.5%)	
					906(99.6%)
	CL1	CL2	CL3	CL4	Total

Fig. 6. Confusion matrix (Test): 1 hidden layer neural network

CL1	221(100%)	0	0	0	
CL2	0	227(100%)	0	0	
CL3	0	1	238(98.3%)	3	
CL4	3	0	0	238(98.7%)	
					924(99.2%)
	CL1	CL2	CL3	CL4	Total

Fig. 7. Confusion matrix (Test): 2 hidden layers neural network

IV. Discussion

The recognition rates obtained by the artificial neural networks are better than those obtained for the SVM. To improve these rates, the time sequences were lowed from 10s to 2s. Compared with the human neural networks, an artificial neural network has to recognize quickly a short video sequence.

Two artificial neural networks were presented, the confusion matrices (Fig. 6 and 7) show that the network with 1 hidden layer has a more interesting recognition rate. On the other hand, the convergence speed is better for the 2 hidden layers network as showed in Fig. 8 and 9.

Columns 1	through 10								
0.0557	0.0634	0.0632	0.2079	0.2626	0.4146	0.2559	0.2018	0.1053	0.3080
0.9921	0.9675	0.9458	0.9025	0.8802	0.8144	0.7016	0.8581	0.9086	0.8406
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0006	0.0008	0.0011	0.0005	0.0004	0.0003	0.0005	0.0006	0.0008	0.0004
...									
Columns 31	through 40								
0.9937	0.9915	0.9875	0.9809	0.9663	0.9741	0.9951	0.9920	0.9786	0.9928
0.0039	0.0144	0.0263	0.0411	0.2021	0.4991	0.1659	0.2106	0.4125	0.1848
0.0001	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0001	0.0001	0.0001	0.0002	0.0001	0.0000	0.0000	0.0000	0.0001	0.0000
...									
Columns 51	through 60								
1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
0.0113	0.0075	0.0072	0.0061	0.0057	0.0039	0.0039	0.0023	0.0029	0.0057
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000

Fig. 8. Example of Walking motion with 1 hidden layer. Each row represent a frame and each line represent the motion classes (line 1: walk, line 2: arms, line 3: legs, line 4: arms/legs).

Columns 1	through 10								
0.9983	0.9990	0.9998	0.9999	1.0000	1.0000	1.0000	0.9999	0.9999	0.9999
0.0695	0.0546	0.0133	0.0050	0.0056	0.0025	0.0017	0.0037	0.0049	0.0039
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0001	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001
...									
Columns 31	through 40								
0.9997	0.9999	0.9998	0.9996	0.9996	0.9996	0.9999	0.9999	0.9996	0.9998
0.0001	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0014	0.0054	0.0144	0.0297	0.0234	0.0121	0.0011	0.0016	0.0004	0.0001
...									
Columns 51	through 60								
0.9999	0.9999	0.9998	0.9999	0.9999	0.9999	0.9999	1.0000	1.0000	0.9999
0.0001	0.0001	0.0005	0.0001	0.0003	0.0001	0.0001	0.0000	0.0000	0.0002
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000

Fig. 9. Example of Walking motion with 2 hidden layers. Each row represent a frame and each line represent the motion classes (line 1: walk, line 2: arms, line 3: legs, line 4: arms/legs).

In [18], the authors propose for the human motion recognition in videos a method which is based on localized space-time features and using a SVM algorithm. For the walking motion (Table I), 83.8% recognition was found and for the waving hand, 73.6% recognition was found (which is assimilated to our hand movements). In our case, 82% of recognition is found for the walking motion and for the arms motion, 96% of detection is found using SVM. The neural networks provided 100% of detection for both recognition.

Zhang et al. [3] use a 4-dimensional local spatio-temporal feature that combines both intensity and depth information. Latent Dirichlet allocation with Gibbs sampling was used as the classifier. For the walking motion, 92% recognition was found and for the waving hand, 95% recognition was found.

Xia et al. [5] present an approach that modelize the joint by using the depth maps. The prototypical poses of actions were found by the reprojected action depth sequences using LDA and then clustered into k posture visual words. With this approach, the recognition reach 96,5% for the walk motion and 100% for the wave motion.

	Walking	Waving hand
Schuld et al.	83,3%	73,6%
Zhang et al.	92%	95%
Xia et al.	96,5%	100%

TABLE I
COMPARISON OF HUMAN MOTION RECOGNITION. SOURCE: SCHULD ET AL. [18], ZHANG ET AL. [3], XIA ET AL. [5].

V. CONCLUSION

In this paper, we wanted to recognize human motion such as walking, arms, legs and arms/legs motion coming from our BVH files data set supplied by the kinect and the Brekel software. We focused on 2 main approaches: support vector machines (SVM) and artificial neural networks (ANN). Our study showed that the ANN provided better results than the SVM. A comparaison has been done using real time and non real time SVM and ANN methods, we could note that the results are better with our methods. For our future works we plan to use the ontology apply to the BVH files and compare to the machine learning algotithm SVM and ANN. We want to improve our data set by including complex motion such as body rotation or half rotation but also discrimate left arm/leg and right arm/leg.

ACKNOWLEDGMENT

The authors would like to thank all the eNTERFACE'12 attendants and the IMS team of Supelec (Metz) who participated in the realisation of a BVH files data set.

REFERENCES

[1] R. Poppe, "A survey on vision-based human action recognition," *Image Vision Comput.*, vol. 28, no. 6, pp. 976–990, 2010.
[2] M. Raptis, D. Kirovski, and H. Hoppe, "Real-time classification of dance gestures from skeleton animation," in *Proceedings of the 2011 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, ser. SCA '11. New York, NY, USA: ACM, 2011, pp. 147–156.
[3] H. Zhang and L. E. Parker, "4-dimensional local spatio-temporal features for human activity recognition," in *IROS*, 2011, pp. 2044–2049.
[4] Z. Ren, J. Yuan, and Z. Zhang, "Robust hand gesture recognition based on finger-earth mover's distance with a commodity depth camera," in *Proceedings of the 19th ACM international conference on Multimedia*, ser. MM '11. New York, NY, USA: ACM, 2011, pp. 1093–1096.
[5] L. Xia, C.-C. Chen, and J. K. Aggarwal, "View invariant human action recognition using histograms of 3d joints." in *CVPR Workshops*. IEEE, 2012, pp. 20–27.
[6] J. Preis, M. Kessel, M. Werner, and C. Linnhoff-Popien, "Gait recognition with kinect," in *Proceedings of the First Workshop on Kinect in Pervasive Computing*, 2012.
[7] S. Saad, D. De Beul, S. Mahmoudi, and P. Manneback, "An ontology for video human movement representation based on benesh notation." ICMCS, 2012, pp. 77–82.
[8] K. E. Raheb and Y. Ioannidis, "A labanotation based ontology for representing dance movement," in *9th Int'l Gesture Workshop*, May 2011.
[9] S. A. Etemad and A. Arya, "Segmentation and classification of human actions and actor characteristics with 3d motion data," in *IJAIA*, 2011.
[10] F. M. Khan, V. K. Singh, and R. Nevatia, "Simultaneous inference of activity, pose and object," in *Proceedings of the 2012 IEEE Workshop on the Applications of Computer Vision*, ser. WACV '12. Washington, DC, USA: IEEE Computer Society, 2012, pp. 281–288.
[11] J. Gu, X. Ding, S. Wang, and Y. Wu, "Full body tracking-based human action recognition," in *ICPR*, 2008, pp. 1–4.
[12] N. A. Ibraheem and R. Z. Khan, "Article: Survey on various gesture recognition technologies and techniques," *IJCA*, vol. 50, no. 7, pp. 38–44, July 2012, published by Foundation of Computer Science, New York, USA.
[13] R. V. Babu and K. R. Ramakrishnan, "Compressed domain human motion recognition using motion history information," in *ICIP (3)*, 2003, pp. 321–324.

[14] M. Meredith and S. Maddock, "Motion capture file formats explained," 2001, department of Computer Science, University of Sheffield. Technical Report CS-01-11.
[15] N. Cristianini and J. Shawe-Taylor, *An Introduction to Support Vector Machines and Other Kernel-based Learning Methods*, 1st ed. Cambridge University Press, 2000.
[16] C.-C. Chang and C.-J. Lin, "LIBSVM: A library for support vector machines," *ACM Transactions on Intelligent Systems and Technology*, vol. 2, pp. 27:1–27:27, 2011, software available at http://www.csie.ntu.edu.tw/ cjlin/libsvm.
[17] C. M. Bishop, *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2006.
[18] C. Schuldt, I. Laptev, and B. Caputo, "Recognizing human actions: A local svm approach," in *In Proc. ICPR*, 2004, pp. 32–36.

Dominique De Beul holds a computer science engineering degree from the University of Mons in Belgium since june 2010. He is pursuing a PhD thesis in the field of complex human motion recognition such as contemporary dance.

Socially Aware Many-to-Machine Communication

Florian Eyben, Emer Gilmartin, Cyril Joder, Erik Marchi, Christian Munier, Kalin Stefanov,
Felix Weninger, Björn Schuller

***Abstract*—This reports describes the output of the project P5: Socially Aware Many-to-Machine Communication (M2M) at the eNTERFACE'12 workshop. In this project, we designed and implemented a new front-end for handling multi-user interaction in a dialog system. We exploit the Microsoft Kinect device for capturing multimodal input and extract some features describing user and face positions. These data are then analyzed in real-time to robustly detect speech and determine both who is speaking and whether the speech is directed to the system or not. This new front-end is integrated to the SEMAINE (Sustained Emotionally colored Machine-human Interaction using Nonverbal Expression) system. Furthermore, a multimodal corpus has been created, capturing all of the system inputs in two different scenarios involving human-human and human-computer interaction.**

I. Introduction

SOCIAL competence, i.e., the ability to permanently analyze and re-assess dialogue partners with respect to their traits (e.g., personality or age) and states (e.g., emotion or sleepiness), and to react accordingly (by adjusting the discourse strategy, or aligning to the dialogue partner) remains one key feature of human communication that is not found in most of today's technical systems. Hence, the SEMAINE project[1] (Sustained Emotionally colored Machine-human Interaction using Nonverbal Expression) built the world's first fully automatic dialogue system with 'socio-emotional skills' realized through signal processing and machine learning techniques. It is capable of keeping sustained conversations with the user, using very shallow language understanding basically, reacting to emotional keywords and allowing simple dialogue acts yet advanced techniques for recognition of affect and non-linguistic vocalizations.

Still, the system is limited to interaction with a single user however, in many real-world scenarios, human-computer interaction with multiple users, and hence, recognizing traits (e.g., personality) and affect-related states (e.g., interest) of the individuals and of the group as a whole, is desirable. Such scenarios include emotional agents incorporated into robots acting as museum guides, or information kiosks. Yet, the generalization from 1 to N system users comes with a variety of 'grand challenges' the following is to be understood as a non-exhaustive list, reaching from front-end to back-end:

1) Speech source localization. Among other applications, this is useful for feedback, such as the avatar / robot turning its head to the person speaking.
2) Technical robustness to non-stationary background noise (transient noise, background speakers) and reverberation in real-world hands-free application scenarios (such as trade fairs, museums etc.)
3) Speaker diarization. This is required for the character to access the interaction history with individual speakers. For instance, it can be used to detect that a person has not been speaking for a longer time; the main challenge is handling overlap between speakers.
4) Even in case of perfect speech detection and absence of overlap or background noise, speech may not be addressed to the virtual agent, but to other humans (side talk), or simply to the speaker itself (self directed talk). This can easily lead to erroneous actions taken by the system.
5) Multi-talker recognition of affect and speech from cross-talk, i. e., in case that system users are speaking simultaneously.
6) Appropriate strategies for dialogue management and adaptation of visual agent behavior, such as 'integrating' users showing a low level of interest while preserving high levels of interest of other users.

Clearly, addressing all these challenges and implementing solutions was beyond the scope of a four week targeted research project. Hence, the M2M project (socially aware Many-to-Machine communication) has focused on some aspects of 2 through 4 in the above list. Precisely, we extended the capabilities of the SEMAINE system to cope with a hands-free scenario where multiple users interact with the system in the presence of background talkers, environmental noise and reverberation, yet assuming little to no overlap between the user utterances targeted to the system. We took advantage of the Microsoft Kinect device[2] for capturing multimodal input and performing some low-level signal treatments. In the result, the SEMAINE system exploits visual as well as audio cues to detect the presence of one or several users and to attribute each utterance to one of them. Furthermore, 'off-talk' utterances, i.e. utterances which are not directed to the system, are detected. Finally, a corpus of multi-user human-human and human-system scenarios have been recorded, to assess the performance of speaker diarization and off-talk detection systems.

The rest of this report is organized as follows: Background work on emotional virtual agent is introduced in Section II. Section III presents an overview of the system resulting from the project. Then, the main functionalities implemented are detailed in Sections IV through VII. The recorded corpus is described in Section VIII, before some conclusions are drawn

Florian Eyben, Cyril Joder, Erik Marchi, Felix Weninger and Björn Schuller are with the Technische Universität München, Germany
Emer Gilmartin is with the Trinity College Dublin, Ireland
Christian Munier is with the Universität Bielefeld, Germany
Kalin Stefanov is with the KTH Royal Institute of Technology, Stockholm, Sweden

[1]http://www.semaine-project.eu/

[2]www.xbox.com/kinect/

in Section IX.

II. Background Work

Aiming to make interaction with virtual agents more natural, a lot of research effort has been invested to equip dialogue systems with social capabilities that go beyond simple verbal skills. These capabilities include aspects of communication that are emotion-related and non-verbal [1]. So far, most systems are tailored for a one-to-one dialogue situation in which one user has a conversation with one virtual agent. Besides purely speech-based systems, also multimodal frameworks considering for example head movements and facial expressions are becoming popular. The SEMAINE system is one example for a (non-task-oriented) multimodal dialogue system that is sensitive to the user's emotion, non-verbal behavior, and affective cues, trying to recognize the user's state and react to it appropriately via multimodal backchannels [2] and feedback [3]. This also includes natural listener behavior such as head nods, smiles, or short vocalizations such as uh-huh or wow. Further, the agent has to determine when to 'take the turn' [4] and produce utterances that fit the dialog context. The 'Sensitive Artificial Listener' scenario used in the SEMAINE system [5], [6] involves four different virtual characters, each of them representing a different emotional state, i.e., a different quadrant in the valence-arousal space. The virtual agents try to induce 'their' emotion in the user, meaning that they have to recognize and display affect. Emotion recognition in multimodal systems is usually based on low-level features characterizing the user's voice, head movements, and facial expression [7]–[9]. As a first step for speech feature generation, voice activity detection has to be applied in order to extract meaningful acoustic features only in regions where the user is talking. In most speech-based emotion recognition engines, features are generated by applying statistical functionals to contours of acoustic low-level descriptors. Among the low-level descriptors are commonly used features such as loudness, fundamental frequency, probability of voicing, Mel-Frequency Cepstral Coefficients (MFCC), and other features based on the signal spectrum [10]. The functionals include common statistical descriptors such as mean, standard deviation, and other analytical descriptors. Automatic speech recognition (ASR) and keyword spotting systems employed for natural human-machine dialog situations have to be noise robust and tailored for spontaneous and emotional speech containing non-linguistic vocalizations such as laughter, sighing, breathing, etc. [11], [12]. These requirements have motivated a lot of research investigating novel speech recognition approaches that go beyond standard hidden Markov modeling [13], [14]. In order to enable combined acoustic and linguistic emotion recognition, bag-of-words features can be computed from the ASR output [15]. As an alternative to categorical emotion recognition based on classes such as 'happiness', 'anger', 'boredom' etc., emotions can also be modeled in a dimensional way by using a continuous scale for affective dimensions like arousal, valence, expectation, intensity, and power in combination with regression techniques such as Support Vector Regression or neural networks with regression outputs [16].

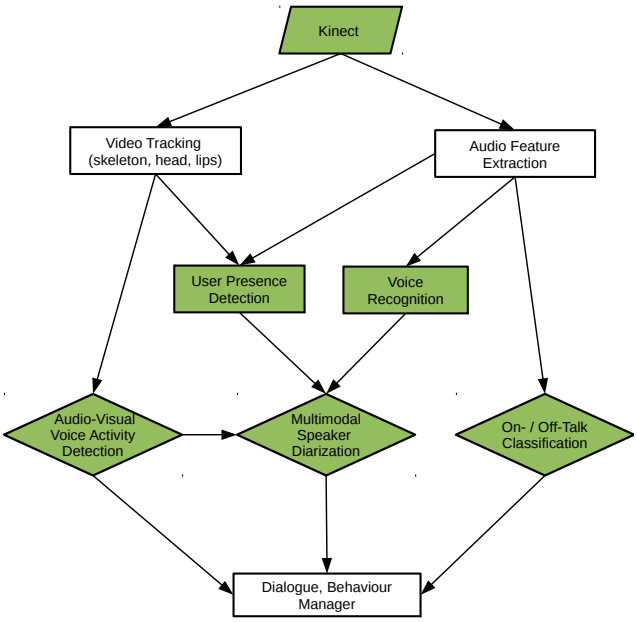


Fig. 1. Overview of the system: simplified flowchart. The colored components are the new modules created in the project.

As emotion tends to evolve slowly over time, context-sensitive classification or regression frameworks that model the evolution of emotion usually prevail over static approaches [17], [18]. In addition to typical emotions or affective dimensions, a socially competent virtual agent also profits from recognizing and reacting to user-specific traits such as personality [19] and states such as the perceived 'level of interest' or paralinguistic information like age and gender [20], [21]. In a hands-free interaction scenario, the influence of reverberation and background noise on ASR and affect recognition has to be taken into account [22], [23]. Furthermore, recognition of the traits and affective states of multiple users has rarely, if ever, been investigated, despite the progress in speaker diarization [24].

III. Overview of the System

A simplified flowchart of the system resulting from the M2M project is represented in Figure 1. As already mentioned, the M2M project has focused on the front-end part of a virtual conversational agent system, in order to extend the possibilities of the existing SEMAINE system for multi-user interaction.

The capabilities of this new front-end are manifold, and comprise:

1) capture of the multimodal input through the Kinect device
2) audio-visual speaker detection and localization
3) audio-visual voice activity detection (VAD)
4) speaker diarization
5) audio off-talk detection.

The 1-st element captures all the input modalities of the Kinect. It consists of three raw sensor inputs (color video, depth video and multi-channel sound), as well as some other

data streams extracted from the raw data (skeleton joints, facial points and enhanced single-channel sound with the source Direction Of Arrival (DOA)).

The 2-nd module relies on the detected user skeletons to determine if one or several users are present. This information is then used by the following elements. The speaker localization is then performed by matching the sound DOA to one of the detected users. The audio-visual VAD component combines the existing audio-only VAD with visual information about the position and movements of the users.

The 4-th item is performed by an online audio speaker diarization algorithm, which can be controlled by the speaker position information. Finally, off-talk detection is achieved by an audio classification approach, analyzing short time-windows of the audio input.

The described elements have been implemented in the C++ language, as components of the openSMILE software [25].

openSMILE is a flexible toolkit for on-line and off-line audio data analysis. It's main purpose is audio feature extraction for speech recognition, paralinguistic audio analysis, and music information retrieval. Components can exchange data via an efficient shared memory component, enabling re-use and sharing of data and a flexible data-flow between components. Many components are currently included for data I/O, acoustic signal processing, extraction of acoustic low-level descriptors, computation of statistical functionals, and classification. The toolkit contains components to read and write data to various common file formats, such as CSV, Weka ARFF format, and Hidden Markov Toolkit (HTK) binary data format. Acoustic descriptors include energy, pitch, voice quality parameters, cepstral coefficients, linear predictive (LP) coefficients, spectral descriptors such as flux, semi-tone chroma and CENS, etc. Further, a set of classifiers including Support-Vector Machines (LibSVM), Long Short-Term Memory Recurrent Neural Networks (LSTM-RNN), and Hidden Markov-Models (HMM) is contained as well as voice activity and turn detector components. Basic networking functionality is included, which allows for receiving audio input over a network and sending back classification results or acoustic descriptors over the network.

IV. Kinect Input Capture and Low-Level Processing

A Kinect for Windows device contains the following hardware components

- Microphone array
- Color sensor
- IR emitter
- IR depth sensor
- Tilt motor

The data generated by the device is accessed through the Kinect for Windows SDK that provides tools and APIs for development of Kinect applications for Microsoft Windows. The following subsections give a brief outline of the raw data used throughout the project.

A. Color and Depth Video

The color data is available at two quality levels and in two different formats. The quality level determines the rate of data transfer from the sensor where the color format determines how the streamed color data is encoded (RGB or YUV). Kinect can stream color data with maximum frame rate of 30 frames per second. The resolution of the color stream is dependent on the frame rate and is specified by the `ColorImageFormat Enumeration`. For example, Kinect can stream raw YUV data with resolution of 640×480 pixels with frame rate of 15 frames per second.

The depth data stream is composed of pixels which contain the distance (in millimeters) from the nearest object to the camera plane at that particular coordinate of the depth sensor's field of view. The resolution of the depth stream is also dependent on the frame rate and is specified by the `DepthImageFormat Enumeration`.

This project uses RGB color data that is streamed by the Kinect device with resolution of 640×480 pixels with frame rate of 30 frames per second. The used depth data is streamed with resolution of 320×240 pixels with frame rate of 30 frames per second.

B. Skeleton and Face Tracking

The device can process the color and depth data to identify up to six human figures in a segmentation map. The segmentation map is a bitmap with pixel values corresponding to the index of the person in the field of view, who is closest to the camera at that pixel position. Up to two of these figures can be fully tracked meaning that the device can locate the joints of the tracked users in space and track their movements over time (called skeleton tracking). Furthermore, the Kinect skeleton tracking is optimized to recognize users that are standing or sitting, and facing the device.

The Face Tracking SDK together with the Kinect for Windows SDK enables tracking of human faces in real time. The face tracking engine analyzes input from a Kinect camera, and can deduce the head pose and facial expressions in real time. The face tracking engine tracks 100 facial points (eye brows, eye contours, nose contour, lip contours, etc.) and can return the location of these points in the 2D coordinate space of the color image. Additionally, the engine tracks the 3D head position and can deduce the pitch, roll, and yaw angles of the head in real time. Finally, the results returned by the engine are also expressed in terms of weights of 6 action units (AUs) and 11 shape units (SUs), which are a subset of what is defined in the Candide3 model[3].

C. Microphone Array and Beamforming

The Kinect has a four channel microphone array. The exact position of the microphones is not specified, but a rough idea can be obtained from the 'specification' at the Microsoft Developer Network[4], indicating that the microphones are placed with 'logarithmic' spacing. The audio input is sampled at

[3] http://www.icg.isy.liu.se/candide/

[4] http://msdn.microsoft.com/en-us/library/jj131033.aspx

16 kHz with 24-bit pulse code modulation (PCM). The Kinect device performs beamforming, source localization, echo cancellation and noise reduction on its own DSP. 11 fixed beams are available, which range from -50 to +50 degrees in 10 degree increments. The API allows selection of a fixed beam, automatic selection of the optimal beam, or simply using the microphones to record 4-channel audio. In the M2M dialogue system, the automatic beam selection is used, where as for data collection purposes in the project, also the 4-channel audio was recorded. Note that it is *not* possible to record the data from all 11 beams at once, probably due to limitations of the DSP's computing power. Source localization estimates the source direction along with a confidence in the interval [0, 1]. According to the specification[5], users should be positioned approximately one to three meters from the microphone array, and noise cancellation and suppression typically provide 30 decibels or more of noise reduction.

D. Integration

For integrating the Kinect input into the SEMAINE system, it was necessary to extend the openSMILE toolkit. To this end, an openSMILE input component was developed. The integration of the audio input is straightforward and can be implemented simply as a replacement for the standard microphone input component, delivering the current beam, the individual microphone signals, or both, as input to the data memory for further processing. Acoustic echo cancellation and automatic noise suppression can be simply switched on or off by means of configuration options. Due to the huge data rate of the RBG and depth images (several gigabytes per minute), their transmission in the openSMILE processing chain was found to create too much overhead. Thus, these data are simply dumped to files on the hard drive.

V. Audio-Visual Scene Model

In order to keep track of different users that interact with the system simultaneously it is crucial to provide a scene model which integrates sensor inputs from different modalities (audio and vision). This enables the correct attribution of utterances to their corresponding user representation in the model. Moreover, fusion of both modalities allows for deciding if there is currently a user talking to the system (*on-talk*) or if users are talking to one another or to themselves (*off-talk*).

A. User Presence Detection

The model represents users by a skeleton tracked via the Kinect device. The information provided through the skeleton tracking comprises the position in the scene and orientation of individual skeleton bones. Due to device capabilities tracking is limited to six simultaneously present skeletons, thus allowing for six users in the scene. If tracking is lost on a user's skeleton, either due to adverse conditions or if they move out of the field of view, the scene model maintains a hypothesis about the user's supposed state and position.

[5]http://msdn.microsoft.com/en-us/library/jj131026.aspx

The audio source localization of the Kinect device provides an azimuth angle to where the currently dominant audio source is supposedly located and a corresponding confidence measure (*audio source confidence*). The user in the scene model that has the smallest angular distance to the supposed audio source location is selected as current speaker. A corresponding confidence measure (*user presence confidence*) is then calculated from the angular distance of the visually selected user to the audio source, the proximity of the selected user to the closest other user in the scene model and from the Kinect's audio source confidence.

B. Audio-Visual Voice Activity Detection

To decide if there is currently a user talking to the system our scene model integrates cues from both the acoustic and the visual modality. The usual approach in speech recognition is to use an energy-based acoustic voice activity detection (VAD) based on the speech signal. Here, we use the acoustic VAD from the SEMAINE system.

In addition to the skeleton data each user is associated with face information from the face tracker if available. The information provided comprises head orientation and animation unit (AU) coefficients. The animation units describe the displacement of certain facial features, as for example the lips and the jaw. Knowing the head orientation it is possible to hypothesize whether a user is currently looking at or away from the system. From the animation units an estimate of the current mouth aperture is computed and integrated over a period of previous frames, yielding a confidence measure for the user's mouth movement.

The overall confidence measure for audio-visual voice activity is then composed from the acoustic voice activity detection, the user presence confidence, the confidence for the user looking at the system and the confidence for movement of the user's mouth.

VI. Speaker Diarization

The speaker diarization module has to identify in real-time who is speaking. One of the difficulty of this task is that no initial information about the speakers is available. In particular, the identity and number of the possible speakers are unknown and the corresponding models have to be built "on the fly". In order to build this system, we first implemented an audio-only system, based on a state-of-the-art algorithm, and then extended it to integrate the additional information from the audio-visual scene model.

A. Audio-Only Speaker Diarization System

The audio speaker diarization system is based on the algorithm proposed by [26]. It is an on-line approach, which allows for a real-time identification of the current speaker, under the assumption that no cross-talk, i. e. at most one person is speaking at a time. The principle of the algorithm is as follows:

- The system is initialized with *general speech models* (one male, one speaker), which play the role of universal background models (UBM), as well as a noise model.

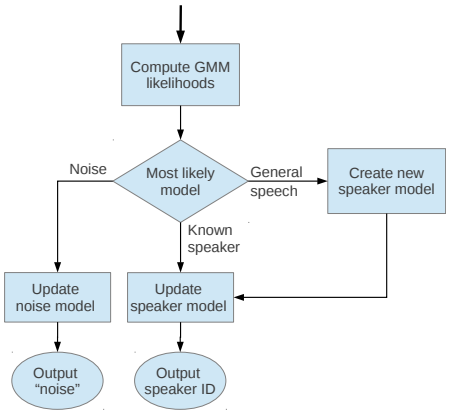


Fig. 2. Flowchart of the audio speaker diarization algorithm.

- At run time, the audio input is continuously compared to the current models. If the most likely model is a general speech model, a new speaker model is created by duplicating the corresponding general model. The selected model is then adapted to the actual observations.

This principle is illustrated by Figure 2.

As in [26], we use MFCC as features describing the instantaneous spectrum and the speaker models are Gaussian Markov Models (GMM), which are updated after the Maximum A Posterior (MAP) adaptation scheme. The considered audio segments consist of a 1 s buffer (100 frames).

The initial models have been learned using parts of our dedicated recorded corpus, which was recorded in the same condition as the use-case scenarios. This corpus is described in Section VIII. The training data was composed of spontaneous speech drawn from two trials of three-person conversations (2 male and 1 female in each trial), yielding about 19 min of audio.

B. Integration of the Audio-Visual Scene Model

We extended the audio-only model of [26], in order to take advantage of the audio-visual scene model described in Section V. We then exploit the voice activity detector (VAD) output, the *user ID*, i. e. the index of the detected user, and the *user presence confidence*, which are the outputs of the user presence detection module.

The flowchart of the modified audio-visual speaker diarization system is display in Figure 3. In this system, we assume that the audio-visual cues are more reliable than the audio-only ones. Thus, they are given the priority on the decision process and the result of the audio algorithm is not fully trusted. Hence, the audio models are used to identify the speaker only when the *user presence confidence* is low (with respect to a certain threshold). Furthermore, new audio speaker models are created only when a new speaker is detected with high confidence by the audio-visual scene model.

Thus, three cases can arise, depending on the result of the audio-visual module:

- **No speech is detected:** the diarization system output the "noise" source and updates the audio noise model.

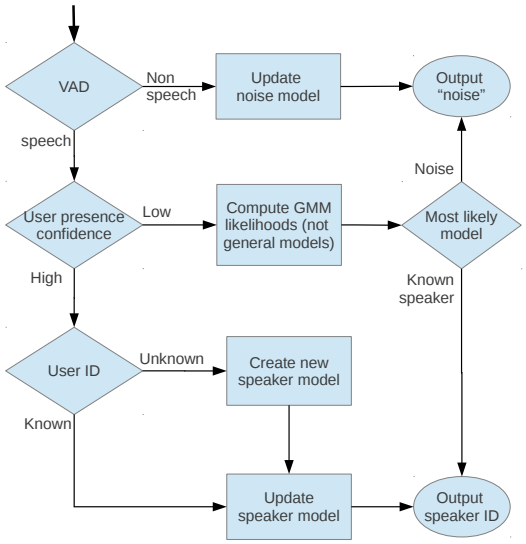


Fig. 3. Flowchart of the audio-video speaker diarization algorithm.

- **A user is detected with high confidence:** the diarization system follows this decision. If no GMM is present for this user, a new one is created by duplicating the most likely *general* model. Then, the speaker model is adapted.
- **A user is detected with low confidence:** the *user ID* is not taken into account and the diarization system relies on the audio information only. The output then corresponds to the most likely GMM. However, no new speaker model can be created and the GMM are not adapted. Hence, we prevent the creation or the update of wrong models, which could be detrimental to the identification accuracy.

VII. Audio Off-Talk Detection

Even though the user presence, head orientation and audio-visual voice activity detection are well able at robustly identifying people talking in front of and directed to the system, there might still be cases where people are not addressing the system but talking to other people, although they are looking at the system. Previous studies [27], [28] suggest that people change their speaking styles depending on their dialogue partners and that this change is detectable by machines. Thus, we can discriminate between speech directed to the system and speech directed to a fellow person by analysing spectral and prosodic characteristics of the speech.

Adopting a similar approach as in [27], we build an on-/off-talk classifier component. The component uses a large set of acoustic features composed of statistics of a broad set of acoustic low-level descriptors and classifies short segments of speech with linear kernel Support-Vector Machines (SVM). The basic idea of this component is, that it shall give further evidence in addition to the multi-modal voice-activity and user presence detection. This evidence will come with a certain lag because a segment of sufficient length (typically 2–5 seconds) is required for analysis. Therefore, the system will process the incoming speech normally, i. e., do multi-modal VAD, perform acoustic analysis and emotion recognition, and

cl. as →	off	on
off	588	53
on	62	166

TABLE I
CONFUSION MATRIX FOR 5-FOLD SCV EVALUATION OF THE ON-/OFF-TALK DETECTOR ON THE EVEN TRIALS OF THE ENTERFACE WORKSHOP SYSTEM INTERACTION DATABASE.

do keyword spotting. Only at the stage of interpretation and agent response generation, the off-talk detection result shall be considered. If the input was classified as being off-talk with a high confidence, or multiple segments of the input are agreeably predicted as off-talk, no agent response shall be prepared and the user state will not be updated with the current results - or, if the user state has already been updated, it shall be reversed. We would like to note at this point, though, that only the off-talk classifier has been implemented during the eNTERFACE workshop, but no modification to the agent behaviour has been implemented. The behaviour suggested in this section, needs to be implemented into the SEMAINE Java components.

The acoustic features used for the off-talk classifier are the same acoustic features that are used in the SEMAINE system for acoustic emotion recognition. This has been chosen for performance and simplicity reasons in this first prototype implementation. Further studies are required to select the most relevant features, and justify a significant overhead by extracting more, or different features in parallel to those used for emotion recognition. 1.882 features are extracted from overlapping fixed length segments of max. 5 seconds sampled at a rate of 1 second (the segments are taken from user speech turns - thus if the turn is shorter than 5 seconds or the total length is not an exact multiple of 5, the last segment might be shorter than 5 seconds).

With the data collected at eNTERFACE, a pre-evaluation experiment was performed for the off-talk detector. 869 instances (fixed length segments - cf. previous paragraph) from the even numbered trials (2, 4, 6, 8) in the database are used for this evaluation; 641 off-talk and 228 on-talk instances. Using a linear SVM with complexity $C = 0.1$ for the Sequential Minimal Optimization (SMO) training algorithm, an accuracy of 86.8 % can be reported for 5-fold stratified cross-validation (SCV). The confusion matrix is shown in table I. This shows the feasibility of the approach, at least for the system demo scenario that was assumed while recording the database. However, we must note, that the folds are not speaker disjunctive due to the random, stratified 5-fold split.

VIII. CORPUS: MULTI-USER INTERACTION

In order to provide training and testing data, a corpus of on-talk and off-talk was collected during the workshop. Speech types of interest were on-talk - human speech directed at a computer, and off-talk - speech between humans in the vicinity of the computer. The corpus comprises samples of speech directed at the computer from different angles (on-talk), and conversational human-human speech recorded at various distances and angles from the computer (off-talk). In all, 11 participants were recorded over 10 sessions, resulting in the collection of over 1.5 hours of recordings.

A. Recording Conditions

The recording setup was designed to allow the collection of examples of on and off talk from different speakers at different angles and distances from the Kinect. Speakers would interact with one another and with the SEMAINE system.

To achieve this a recording space was set up around a monitor attached to the computer running SEMAINE. Recordings were made on two Kinect sensors mounted in a vertical stack on top of the 21" flat screen monitor which displayed the SEMAINE system user interface.

The floor of the recording space was marked with three lines radiating from a point directly below the two Kinects. Two of the lines marked the perpendicular and left and right edges of the Kinect's camera's field of view, while the third was perpendicular to the screen. The lines were marked along their length at 80, 160, and 240 cm from the origin. These markings resulted in nine possible locations for a speaker to stand - left, right, and centre at three different distances, and were used to orientate participants during the recordings.

Two scenarios were used for the recordings. In the first, three participants stood at the left, centre, and right of the Kinect's visual range and spoke to one another. Conversation consisted of short casual exchanges arising from questions about everyday topics. Participants could speak about subjects of their choosing or use prompt sheets provided by the experimenters. In this scenario nine exchanges were recorded per session, with participants changing place after each exchange so that samples would be collected from the nine possible positions.

In the second scenario, three participants stood 60-80 cm from the screen and interacted with the SEMAINE system and with each other, changing angular position at intervals to allow samples to be collected for all speakers from all three angles to the machine.

Eleven people participated in the recordings (7 male and 4 females). All speech was in English. All participants regularly used English in their work, and three were native speakers. All participants were familiar with speech technology and had experience of dialogue systems.

The two Kinects were used to collect audio, video and depth images for each session. Recordings were immediately backed up to a separate hard drive.

All of the WAV audio recordings were annotated using Praat [29], in terms of speaker, distance from computer, angular position of speaker (Left, Centre, Right), and type of speech (on-talk, off-talk). The data from the annotated recordings were then used to train and test the speaker diarization system developed in the m2m project.

IX. CONCLUSION

The project "Socially Aware Many-to-Machine Communication" has resulted in the implementation of a new front-end for the SEMAINE dialogue system, which extends its capabilities for multi-user support. The new system takes

advantage of the Microsoft Kinect device for capturing and processing multimodal input in real-time. An interface component has been created for importing the raw input signals (four-channel audio, color video and depth images) as well as the extracted information (enhanced one-channel audio, skeleton and facial points tracking) from the Kinect into the existing openSMILE framework. New algorithms for audio-visual Voice Activity Detection (VAD), multimodal speaker diarization and off-talk detection have been implemented in this framework. The results obtain in preliminary experiments indicate the potential of our approach for the handling of a multi-user scenario. Furthermore, a corpus of multimodal spontaneous interaction recordings has been collected. The data comprise all the modalities captured by the Kinect, and correspond to two scenarios of human-human and human-machine dialogues.

This project opens the path to several possible future working directions. First, the implementation of new behaviours which would take into account the multi-user scenarios (such as turning the "face" to the current speaker or addressing a specific user) is now possible, thanks to the delivered front-end. The design of data-driven algorithms for the fusion of multimodal information can also be considered, instead of the heuristic rule-based approaches followed in the VAD and speaker diarization components. Finally, an open-source version of the created software will be released in the near future.

REFERENCES

[1] R. Cowie, "Describing the forms of emotional colouring that pervade everyday life," in *The Oxford Handbook of Philosophy of Emotion*, P. Goldie, Ed. Oxford University Press, 2010, ch. 3, pp. 63–94.

[2] V. H. Yngve, "On getting a word in edgewise," in *CLS-70*. University of Chicago, 1970, pp. 567–577.

[3] J. Allwood, J. Nivre, and E. Ahlsén, "On the semantics and pragmatics of linguistic feedback," *Journal of Semantics*, vol. 9, no. 1, pp. 1–26, 1992.

[4] H. Sacks, E. Schegloff, and G. Jefferson, "A simplest systematics for the organization of turn-taking for conversation," *Language*, vol. 50, pp. 696–735, 1974.

[5] M. Schröder, R. Cowie, D. Heylen, M. Pantic, C. Pelachaud, and B. Schuller, "Towards responsive Sensitive Artificial Listeners," in *Proceedings 4th International Workshop on Human-Computer Conversation*, Bellagio, Italy, October 2008, 6 pages.

[6] M. Schröder, E. Bevacqua, R. Cowie, F. Eyben, H. Gunes, D. Heylen, M. ter Maat, G. McKeown, S. Pammi, M. Pantic, C. Pelachaud, B. Schuller, E. de Sevin, M. Valstar, and M. Wöllmer, "Building Autonomous Sensitive Artificial Listeners," *IEEE Transactions on Affective Computing*, 2012, 20 pages, to appear.

[7] B. Schuller, A. Batliner, S. Steidl, and D. Seppi, "Recognising Realistic Emotions and Affect in Speech: State of the Art and Lessons Learnt from the First Challenge," *Speech Communication, Special Issue on Sensing Emotion and Affect - Facing Realism in Speech Processing*, vol. 53, no. 9/10, pp. 1062–1087, November/December 2011.

[8] H. Gunes, B. Schuller, M. Pantic, and R. Cowie, "Emotion Representation, Analysis and Synthesis in Continuous Space: A Survey," in *Proceedings International Workshop on Emotion Synthesis, rePresentation, and Analysis in Continuous spacE, EmoSPACE 2011, held in conjunction with the 9th IEEE International Conference on Automatic Face & Gesture Recognition and Workshops, FG 2011*, IEEE. Santa Barbara, CA: IEEE, March 2011, pp. 827–834.

[9] M. F. Valstar, B. Jiang, M. Mehu, M. Pantic, and K. R. Scherer, "The first facial expression recognition and analysis challenge," in *FG*, 2011, pp. 921–926.

[10] B. Schuller, M. Wöllmer, F. Eyben, and G. Rigoll, "Spectral or Voice Quality? Feature Type Relevance for the Discrimination of Emotion Pairs," in *The Role of Prosody in Affective Speech*, ser. Linguistic Insights, Studies in Language and Communication, S. Hancil, Ed. Peter Lang Publishing Group, 2009, vol. 97, pp. 285–307.

[11] M. Wöllmer, F. Eyben, J. Keshet, A. Graves, B. Schuller, and G. Rigoll, "Robust Discriminative Keyword Spotting for Emotionally Colored Spontaneous Speech Using Bidirectional LSTM Networks," in *Proceedings 34th IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP 2009*, IEEE. Taipei, Taiwan: IEEE, April 2009, pp. 3949–3952.

[12] M. Wöllmer, F. Eyben, A. Graves, B. Schuller, and G. Rigoll, "Bidirectional LSTM Networks for Context-Sensitive Keyword Detection in a Cognitive Virtual Agent Framework," *Cognitive Computation, Special Issue on Non-Linear and Non-Conventional Speech Processing*, vol. 2, no. 3, pp. 180–190, 2010.

[13] M. Wöllmer, F. Eyben, B. Schuller, and G. Rigoll, "A Multi-Stream ASR Framework for BLSTM Modeling of Conversational Speech," in *Proceedings 36th IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP 2011*, IEEE. Prague, Czech Republic: IEEE, May 2011, pp. 4860–4863.

[14] M. Wöllmer, B. Schuller, and G. Rigoll, "A Novel Bottleneck-BLSTM Front-End for Feature-Level Context Modeling in Conversational Speech Recognition," in *Proceedings 12th Biannual IEEE Automatic Speech Recognition and Understanding Workshop, ASRU 2011*, IEEE. Big Island, HY: IEEE, December 2011, 6 pages, to appear.

[15] F. Eyben, M. Wöllmer, A. Graves, B. Schuller, E. Douglas-Cowie, and R. Cowie, "On-line Emotion Recognition in a 3-D Activation-Valence-Time Continuum using Acoustic and Linguistic Cues," *Journal on Multimodal User Interfaces, Special Issue on Real-Time Affect Analysis and Interpretation: Closing the Affective Loop in Virtual Agents and Robots*, vol. 3, no. 1–2, pp. 7–12, March 2010.

[16] F. Eyben, M. Wöllmer, and B. Schuller, "A Multi-Task Approach to Continuous Five-Dimensional Affect Sensing in Natural Speech," *ACM Transactions on Interactive Intelligent Systems, Special Issue on Affective Interaction in Natural Environments*, 2012, 29 pages, to appear.

[17] M. Wöllmer, B. Schuller, F. Eyben, and G. Rigoll, "Combining Long Short-Term Memory and Dynamic Bayesian Networks for Incremental Emotion-Sensitive Artificial Listening," *IEEE Journal of Selected Topics in Signal Processing, Special Issue on Speech Processing for Natural Interaction with Intelligent Environments*, vol. 4, no. 5, pp. 867–881, October 2010.

[18] A. Metallinou, M. Wöllmer, A. Katsamanis, F. Eyben, B. Schuller, and S. Narayanan, "Context-Sensitive Learning for Enhanced Audiovisual Emotion Classification," *IEEE Transactions on Affective Computing*, 2012, 14 pages, to appear.

[19] G. Mohammadi, A. Vinciarelli, and M. Mortillaro, "The voice of personality: Mapping nonverbal vocal behavior into trait attributions," in *Proceedings of ACM Multimedia Workshop on Social Signal Processing*, Florence, Italy, Oct. 2010, pp. 17–20.

[20] F. Burkhardt, M. Eckert, W. Johannsen, and J. Stegmann, "A database of age and gender annotated telephone speech," in *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*, N. C. C. Chair), K. Choukri, B. Maegaard, J. Mariani, J. Odijk, S. Piperidis, M. Rosner, and D. Tapias, Eds. Valletta, Malta: European Language Resources Association (ELRA), May 2010.

[21] B. Schuller, R. Müller, F. Eyben, J. Gast, B. Hörnler, M. Wöllmer, G. Rigoll, A. Hthker, and H. Konosu, "Being Bored? Recognising Natural Interest by Extensive Audiovisual Integration for Real-Life Application," *Image and Vision Computing, Special Issue on Visual and Multimodal Analysis of Human Spontaneous Behavior*, vol. 27, no. 12, pp. 1760–1774, November 2009.

[22] B. Schuller, "Affective Speaker State Analysis in the Presence of Reverberation," *International Journal of Speech Technology*, vol. 14, no. 2, pp. 77–87, 2011.

[23] M. Wöllmer, F. Weninger, S. Steidl, A. Batliner, and B. Schuller, "Speech-based Non-prototypical Affect Recognition for Child-Robot Interaction in Reverberated Environments," in *Proceedings INTERSPEECH 2011, 12th Annual Conference of the International Speech Communication Association*, ISCA. Florence, Italy: ISCA, August 2011, pp. 3113–3116.

[24] D. A. Reynolds, P. Kenny, and F. Castaldo, "A study of new approaches to speaker diarization," in *INTERSPEECH*, 2009, pp. 1047–1050.

[25] F. Eyben, M. Wöllmer, and B. Schuller, "openSMILE – The Munich Versatile and Fast Open-Source Audio Feature Extractor," in *Proceedings of the 9th ACM International Conference on Multimedia, MM 2010*, ACM. Florence, Italy: ACM, October 2010, pp. 1459–1462.

[26] J. T. Geiger, F. Wallhoff, and G. Rigoll, "Gmm-ubm based open-set online speaker diarization." in *INTERSPEECH*, T. Kobayashi, K. Hirose, and S. Nakamura, Eds. ISCA, 2010, pp. 2330–2333.

[27] A. Batliner, V. Zeissler, E. Nöth, and H. Niemann, "Prosodic classification of offtalk: First experiments," in *Proc. of the 5th International Conference on Text, Speech and Dialogue*. Springer, London, UK, 2008, pp. 357–364.

[28] A. Batliner, B. Schuller, S. Schaeffler, and S. Steidl, "Mothers, adults, children, pets – towards the acoustics of intimacy," in *Proc. of ICASSP 2008, Las Vegas, Nevada, USA*. IEEE, 2008, pp. 4497–4500.

[29] P. Boersma and D. Weenink, "Praat: doing phonetics by computer," [Computer program] http://www.praat.org/, 2012.

Is This Guitar Talking or What !?

M. Astrinaki, L. Reboursiere, A. Moinet, R. Graham, N. d'Alessandro, T. Dutoit

Circuit Theory and Signal Processing Lab, University of Mons, Belgium

{maria.astrinaki, loic.reboursiere, alexis.moinet, nicolas.dalessandro
nicolas.dalessandro, thierry.dutoit}@umons.ac.be, info@rickygraham.net

Abstract

In this project, we explore the possibility for an augmented guitar to be used as a controller for the expressive manipulation of reactive synthetic speech. This idea comes at the intersection of two research frameworks. On the one hand, we aim at augmenting the electric guitar by extracting guitar playing techniques directly from the guitar sound, through an hexaphonic pickup (one microphone per string). On the other hand, we develop the MAGE software, a unique set of tools for generating high-quality HMM-based speech synthesis in a reactive way. Bringing these two technologies together allows us to explore various mappings between the controller and the speech synthesis, and propose expressive solutions.

Index Terms: HMMs, speech synthesis, reactive control

1. Introduction

Speech is one of the richest and most ubiquitous modalities of communication used by human beings. Vocal expression involves complex production and perception mechanisms. Conversation is a highly interactive process, with complex timings and wide-ranging variations of quality. It is known that speech production properties have a deep impact on perceived identity and social cues [1]. This critical role of speech production in our life makes anybody an expert listener. The synthesis of artificial speech has been explored for decades to use in many applications, from the purely functional level to artistic exploration. However, human's natural expertise in listening to spoken content makes speech synthesis a really complex problem. Recent synthesizers have made great progress in terms of intelligibility and naturalness but they are still not providing a completely convincing vocal experience to users, neither an expressive tool for artists. In this Section, we describe the various research problems that lead to this situation, as an introduction to our concept of *tangible speech synthesis* and the new speech synthesis system that we present.

1.0.1. From Speech Production to Social Cues

Understanding voice production requires an interdisciplinary approach. It can be seen bio-mechanically as pulmonary pressure being applied on tensed vocal folds and the manner of placing the various articulators in the vocal tract, such as tongue, jaw or lips [2]. The acoustics of this phenomenon suggest that the volume velocity waveform generated by the vocal folds vibration propagates through pharyngeal, oral and nasal cavities with time-varying resonance frequencies, called *formants* [3]. Linguists are interested in how these formants vary over time and their relation with vocal tract postures that we continuously browse when we speak, called *phonemes* [3]. They also study, what is called *prosody* [3], how fundamental frequency and amplitude of vocal folds vibration vary over time, as well as phoneme durations. Phonetical and neurological studies show that upcoming speech fragments are planned ahead by the brain, and then corrected on-the-fly by continuously evaluating acoustical and sensorial distances from the plan [2].

This active research community has been making outstanding progresses over the last decades, but it seems that some aspects of speech production remain misunderstood. For example, we do not have an exhaustive model for vocal folds vibration, because observations *in vivo* are nearly impossible. There is also a big debate in how speech production is influenced by the context, such as speaker's emotional state, listener's reaction or other surrounding stimuli, because real-life measurements are intrusive. These issues result in an elusive mapping between parameters of existing production models and real social impacts of speech, such as intents or emotions, as illustrated in Figure 1.

1.0.2. Text-To-Speech and User Interaction

Early research in speech synthesis was firstly trying to model the physiology of speech production, then manipulating the models according to what was observed in the vocal tract or on the speech spectrum, such as the well-known source-filter model [4]. The generalization of computing in speech synthesis research progressively benefited to a new approach, detached from physiological roots, and focused on the systematic conversion from text to speech. New algorithms from the early 1990s, based on waveform segmentation and concatenation [5], made a remarkable leap forward in term of intelligibility and

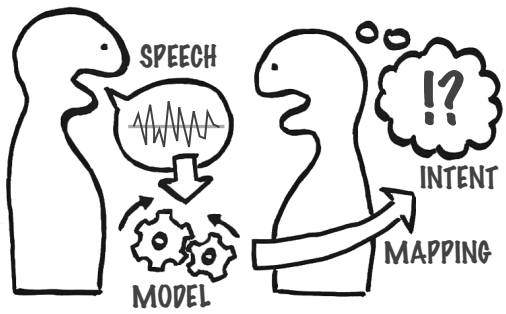


Figure 1: Major obstacles remain in order to accurately map between parameters of speech production models and perceived intents or emotions.

naturalness, shifting the physiological trend to focus on speech simulation.

Text-To-Speech (TTS) systems have a common architecture and work in two steps. First, text is converted into the *narrow phonetic transcription* by the Natural Language Processor (NLP), containing phonemes and prosody. Then this information is converted into speech sound by the Digital Signal Processor (DSP) [3], as illustrated in Figure 2.

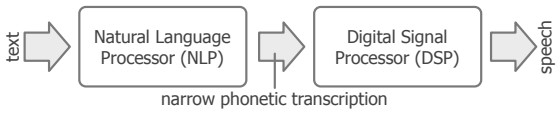


Figure 2: Text-To-Speech: NLP converts text into the narrow phonetic transcription, then DSP converts this transcription into speech sound.

At the time TTS became the main trend, it was not evident that computing would go mobile so massively. Retrospectively, we understand how the design choices underlying TTS – text input and black-boxed generation of resulting speech – have anchored its use to reading text on desktop computers. However, mobile computing relies on ubiquitous sensing of user's context, and user interaction tends to become more natural. Therefore, high-quality speech synthesis seems to have major issues in being used "in the wild". Nowadays there are two main application types that are prevented to expand because of these limitations:

1. *Context-reactive speech synthesis*: our modern life is encountering an increasing amount of virtual agents, on the phone, in the car or in public spaces. Ubiquitous computing brings these systems to gather a lot of information about our context: location, light/noise conditions, movements, social connections, etc. Most of this information is dynamically changing. However, embedded TTS makes few sense of these context changes, because the speech production properties can barely be altered, even less on-the-fly.

2. *Performative speech synthesis*: artificial speech can be generated from gestural performance, rather than pre-typed text. This approach has many applications, such as silent speech communication [6], speech production replacement for voice-impaired users, etc. This technique is also interesting for artistic purposes, as speech is a common medium used in various disciplines. These situations require a major breakthrough in speech synthesis techniques in order to create speech sounds from non-textual fast-changing inputs.

1.1. Guitar as the Controller

One essential aspect involved in developing a system for performative speech synthesis is the design of the controlling device. One approach to this research is to consider that the purpose is to design a new instrument for musical expression, or NIME, here applied to the speech signal. A common concern in NIME research is the lack of human practice associated to new instruments. This issue can trap the design process in a loop where the lack of a good device prevents good practice to appear and the lack of good practice prevents a good device to emerge. In order to avoid such a deadlock situation, many people have started their NIME design from an existing instrument. Indeed the existing practice can be reused and then extended for the new purpose. Due to its wide availability and contemporary history, the electric guitar has been a good candidate for such a strategy [7, 8].

In this project, we have decided to use the electric guitar as the input device for controlling speech synthesis. It is motivated by the above-mentioned intent to reuse and extend the existing guitar playing techniques for our new purpose, but also because we wanted to further explore the Guitar As Controller hardware/software platform that we have built in the lab for the last 2-3 years.

1.2. Outline of the Report

In this report, we start describing the speech synthesis technique we use in this project, called HMM-based synthesis in Section 2. Then we describe the main modifications that we have applied to state-of-the-art HMM-based speech synthesis in order to create a fully reactive and controllable sound synthesis system, called MAGE, in Section 3. Section 4 explains the modifications applied on the existing Guitar As Controller toolbox and new mapping strategies that have been developed specially for controlling speech. Finally we discuss the prototype that we could assemble and test during the workshop in Section 5.

2. HMM-Based Speech Synthesis

Nowadays, the most common approach for achieving high quality natural speech synthesis is the corpus-based unit selection technique. In principle, this method relies on runtime selection and concatenation of speech units from a large speech database using explicit matching criteria [5]. In direct contrast to the dominance of corpus-based unit selection, there is an increased interest for statistical parametric speech synthesis [9]. Statistical parametric speech synthesis is based on an model-based parametric framework, where speech is generated by averaging sets of similarly sounding speech segments. Indeed, instead of using real speech samples at runtime, context-dependent HMMs are trained from the databases of natural speech, and then speech waveforms are generated from the HMMs themselves.

2.1. Core architecture of typical system

In a typical statistical parametric speech synthesis system, the pre-recorded database is analysed, various production parameters are extracted spectral envelopes, fundamental frequency and duration of the phonemes - and used to train statistical models. Usually a maximum likelihood (ML) criterion is used to estimate the model parameters [10]. Later these models will generate the speech parameters for a given targeted text input. Speech waveforms are produced from the parametric representations of speech with typical speech synthesis techniques: subtractive synthesis [11] or harmonic plus noise [12]. Any generative model can be used, however most widely used are Hidden Markov Models (HMMs) [13], and this approach is known as HMM-based speech synthesis [10].

In Figure 3 we present an overview of a typical HMM-based speech synthesis system (HTS) [14], which consists of a training and a synthesis part.

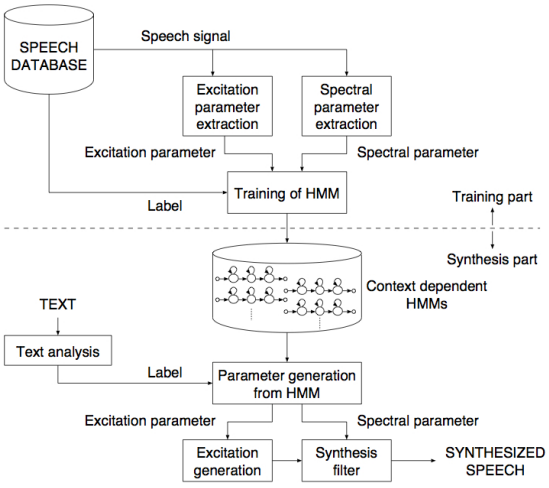


Figure 3: *Overview of a typical HMM-based speech synthesis system (HTS) [14].*

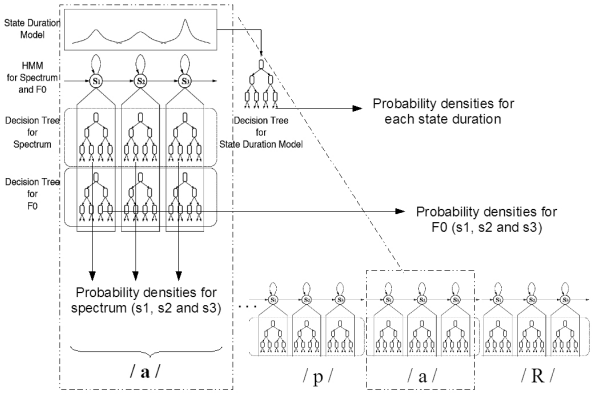


Figure 4: *Decision trees for context clustering [18].*

During the training part both spectrum (mel-cepstral coefficients [15], and their dynamic features) and excitation (logarithmic fundamental frequency (logF0) and its dynamic features) parameters are extracted from a natural speech database. These parameters are then modeled by context-dependent HMMs, taking into account phonetic, linguistic and prosodic contexts. Multi-space probability distributions (MSD) [16] are used to properly model the excitation parameters which is a variable dimensional parameter sequence with non continuous pitch values in unvoiced regions. In order to model speech temporally, HMMs model the state duration densities by using multivariate Gaussian distributions [17]. So to handle all the contextual factors, such as phone identity and stress or accent related factors that affect the targeted synthetic speech output, decision-trees based on context clustering techniques [18] are used, as shown in Figure 4. Magnitude spectrum, fundamental frequency and duration are modeled independently, therefore there is a different phonetic decision tree for each of these features [19].

At the synthesis part, the input text is analyzed and converted to a context-dependent phoneme sequence. Then by concatenating the context-dependent HMMs according to the generated context-dependent phoneme sequence an HMM utterance is constructed. This HMM utterance is used to generate the sequences of spectral and excitation parameters [20], and by using excitation generation and a speech synthesis filter (e.g., mel log spectrum approximation (MLSA) filter [21]) the final speech waveform is reconstructed.

2.2. Advantages

Compared to the unit-selection synthesis, statistical parametric synthesis offers significant advantages not only in controlling the synthesis procedure but also in being much more flexible due to the well defined statistical modeling process. One of its main advantages is the flexibility in changing its voice characteristics and speaking

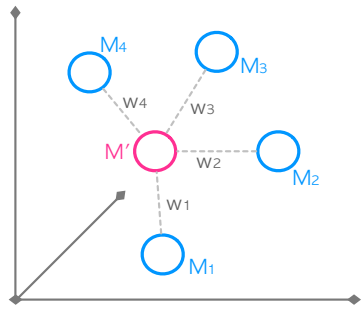


Figure 5: Speakers individuality modeled by HMMs, Mi, where Wi denotes the interpolation weight between the existing models in order to produce the untrained voice characteristics, M'.

styles by simply transforming the model parameters. This is possible through :

- adaptation, mimicking voices by means of maximum a posteriori (MAP) estimation [22] and maximum likelihood linear regression (MLLR) [23].
- interpolation, mixing voices and synthesize speech with untrained voice characteristics [24] as shown in Figure 5.
- eigenvoice, producing new voices, [25].
- multiple regression, to control voice characteristics intuitively [26].

Other advantage is that HTS can easily be adapted in different languages, contexts and applications [14]. Compared with unit-selection synthesis, statistical parametric synthesis has a very small footprint, just a few MBytes [27], since there are no real speech samples used, only the statistics of acoustic models are stored and fewer tuning parameters since both modeling and synthesis processes are based on mathematically well-defined statistical principles. However the main drawbacks of statistical parametric synthesis is the quality of the final synthesized waveform. The reason for this quality degradation seems to be the vocoders used, the acoustic modeling accuracy and finally the over-smoothing [28].

3. Reactive HMM-Based Speech and Singing Synthesis

As the new trends in understanding expressivity in speech are being explored, and the need for real world speech and singing synthesis applications such as entertainment and gaming applications, silent speech communication and performing arts application as well as assistive applications for speech impaired people. However one might notice that a real solid platform for performative speech and singing synthesis is missing. The challenges of such a platform though are on the one hand the reactive production of expressive speech and the adaptability and latency of the speech synthesis and on the other hand how to provide a meaningful gestural control mechanism.

In traditional HMM-based speech synthesis, as described in Section 2, a certain amount of text is required in advance to be processed and converted into speech as a whole target but during this text to speech conversion process any external influence is rather limited. This limitation prevents to adapt to any external solicitation within the sentence as it is being synthesized. Thus, we decided to build MAGE, which proposes the generation of speech parameters within a smaller look-ahead window rather than the whole available text. This approach allows to infer on speech outputs at various production levels and time scales. However, such a system has totally different requirements than the original one; it needs to have a reactive programming architecture, and to be both listener-specific and context-aware. To our current knowledge, MAGE is the first platform for reactive programming of speech and singing synthesis able to address these issues, allowing reactive prosodic and contextual user control.

3.1. Short-Term Speech Parameter Trajectories

As inherited form the original system HTS; MAGE also has a training and a synthesis part. For both systems the training part is identical, as described in [28], but their fundamental difference lays in the synthesis part. During the synthesis time of HTS, the input text is analyzed and converted to a context-dependent phoneme sequence, then according to this sequence, context-dependent HMMs are concatenated, constructing an HMM utterance. Then this HMM utterance is used to generate the sequences of spectral and excitation parameters by maximising the probability of the speech parameter sequence [28]. Consequently, in HTS the smallest accessible time scale is the complete targeted word sequence. Finally, the targeted speech output is reconstructed by using excitation generation and a speech synthesis filter, here Mel Log Spectrum Approximation (MLSA) filter [21] with pulse-train or white-noise excitation.

In direct contrast to HTS synthesis part, in MAGE the observation window is reduced, from all the available phoneme sequence to just a sliding window of two phonemes. More specifically, as the available phonemes are being streamed as input to MAGE, only the new phoneme and the previous phoneme are used to concatenate the context-dependent HMMs and construct an HMM utterance. Then this HMM utterance, consisting only of the context-dependent HMMs of two phonemes is used to generate the corresponding sequences of spectral and excitation parameters. Then, by using the maximisation in Equation 2, as described in [28] the speech parameter trajectories are generated. A result of the reduced ob-

servation window approach is that the generated speech parameter trajectories do not correspond to the overall maximum of probability (HTS), but only the concatenation of locally-maximized speech parameters (MAGE).

$$q^\star = \underset{q}{\operatorname{argmax}}\, P(q \mid \boldsymbol{\lambda}^\star, \hat{\boldsymbol{T}}) \tag{1}$$

$$\hat{\boldsymbol{O}} = \underset{O}{\operatorname{argmax}}\, P(O \mid q^\star, \boldsymbol{\lambda}^\star, \hat{\boldsymbol{T}}) \tag{2}$$

where O and q are respectively the sequence of speech parameters and the sequence of states, $q^\star$ and $\lambda^\star$ respectively the estimated sequence of states and the concatenated left-to-right context-dependent HMMs of the 2-phoneme window, $\hat{O}$ the sequence of locally-maximized generated speech parameters, and $\hat{T}$ is the time frame corresponding to the first label of the 2-phoneme window on which $\hat{O}$ is computed.

By using a 2-phonemes window for the speech parameter trajectory generation, MAGE opens the enclosed synthesis loop of HTS, and the initial accessible time scale of the sentence level in now reduced to the phoneme level. As Figure 6 illustrates, when there are two phonemes for the sliding window, the speech parameter trajectories are generated and the corresponding speech samples are synthesized and stored in independently. By providing the needed real-time audio architecture, as it is described in Section 3.2, the audio samples can be synthesized, altered and streamed on the fly, only with one phoneme delay. In other words, it is possible to influence the generation of the speech parameters and change the corresponding speech samples with a delay of only one phoneme. As follows, the spectral envelopes, the phoneme durations as well as the pitch curves can be modified as speech samples are being synthesized and affect the final output with only one phoneme delay.

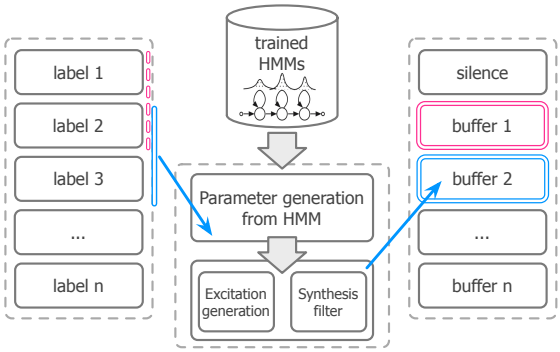


Figure 6: MAGE *synthesis, using a 2-phoneme sliding window to generate the speech parameter trajectories and audio buffers.*

3.2. The MAGE platform

MAGE is a platform for reactive HMM-based speech and singing synthesis. It provides an API for reactive programming in C/C++, aimed at being included in realtime audio softwares. MAGE is thread safe and engine independent. It is the shell that provides to HTS the needed real-time audio architecture so that the targeted speech samples can be reactively manipulated. As illustrated in Figure 7, MAGE consists of the label thread, that control how the inputed phonemes are streamed to be processed, the parameter generation thread, that generates the sequences of spectral and excitation parameters by maximizing the probability of the speech parameter sequence and finally the audio generation thread that generated the targeted speech samples. During runtime MAGE will allow the input of user controls so that the speech samples finally outputted in to the audio thread can be reactively controlled.

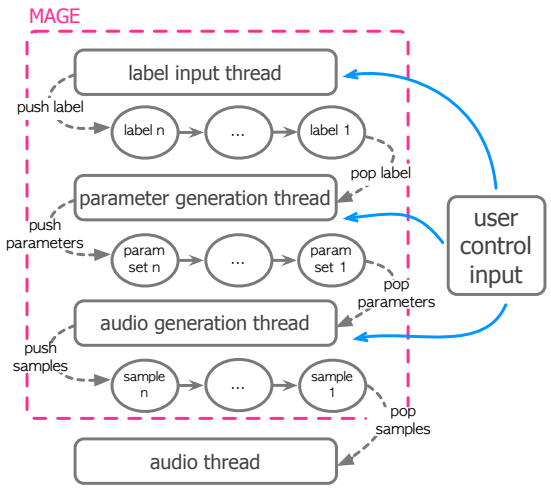


Figure 7: *Multithread architecture of* MAGE*: speech synthesis thread makes the connection between the user control and the audio thread.*

Since MAGE can be easily imported in various frameworks and it can be simply combined with OSC-enabled sensors, it allows fast and easy prototyping. Additionally it provides easy context and prosody controls over the synthesized voice. Contextual control is implemented based on the label thread and how the available phonemes are streamed into MAGE, while prosody control is based on the reactive manipulation of the pitch trajectories, phoneme duration and vocal tract parameter.

MAGE comes as a consequential implementation, following the idea of performative speech synthesis, as a way of looking beyond Text-To-Speech (TTS). It is an interdisciplinary project, addressing problems in the fields of speech processing, linguistics and human-computer interaction (HCI) and it attempts to bring a common platform to address their problems. MAGE is targeted to be used for approaching and understanding longterm ques-

tions in speech production, such as degrees of coarticulation, speech motor control, speech planning, intonation, voice quality, speech time scales, etc. through gestural control and interactive interfaces, mainly through mobile and social computing.

4. The guitar as a controller

4.1. Introduction

The first time time guitars were used as controllers can be correlated to the appearance of MIDI guitars. On those systems, an hexaphonic pickup (1 pickup per string) enables polyphonic pitch and amplitude tracking so as to drive synthesizers or samplers in order to extend the sounding possibilities of the instrument. The guitar became, thus, a MIDI controller.

Later, the augmented instruments term and field appeared pushing further the notion of controller by embracing the more global notion of gesture and more specifically of musical gesture. Anything that gives the user controls on the produced sound is a musical gesture.

In [29], the author defined the three types of musical gestures:

- effective: physical technique employed by the agent to produce sound (picking, fretting, etc.)
- ancillary: accompanying physical body movements
- figurative: note attack, scalar instrumental structures, melodic contour

These three types of musical gestures are as many possible ways to have control on the resulting sound. Regarding the guitar, the three types of musical gesture have been assessed in many different ways expanding each time the controller notion.

Ancillary musical gesture, e.g, have been used and mapped to audio effects in [7] and [8]: in the first example, an inclinometer on the head of the guitar was controlling the volume of the effect. On the second example, pressure sensors have been added to the rear of the guitar to catch information on the movement the guitar does around the belly while the guitarist is playing. This information was then mapped to the different parameters of a wah-wah like effect.

Figurative musical gestures have been used in [30] to create a system which analyses the melodic structure it receives and applies different mappings depending on which pitch contour is detected. Different effective gestures have been analysed and detected in [31] and [32]. In [33], we developed algorithms for most of the major guitar playing techniques, i.e, hammer-on, pull-off, slide, palm muting, bend, natural harmonic notes as well as detection of the plucking point.

During this project, we mainly worked on the first (effective) type of gesture using the algorithms developed in [33] and implemented in Max MSP software. The third type (figurative) of gestures was taken into account, but due to a lack of time this type of gesture was not included in the final mapping.

4.2. Guitar playing techniques detection and optimization

The detection of the playing techniques detection was made possible by the use of an hexaphonic pickup (one pickup per string, i.e six separate signals), e.g [1]Roland GK-3, which implies that six separate analysis had to run at the same time. The Max MSP implementations of the algorithms were first done one by one in order to test separately their real-time efficiency. However, once grouped together and working at the same time, the CPU consumption increased dramatically.

To address this problem of CPU consumption, the first and main step was to gather every detections in one patch to define only the needed FFT. Indeed `fft~` Max MSP object is quite CPU consuming and the first implementations of the algorithms were using nearly 6 FFT for each playing technique. Gathering all the algorithms dropped down the number of FFT at 12, 2 per string: one was from `fft~` Max MSP object and the other one from the `sigmund~` external (third party object) used for the pitch extraction. It has to be noticed that a FFT developed in C language into an external, i.e `sigmund~`, is less consuming than an `fft~` object.

The second element that has been tested was the difference between the use of abstractions (instances of a patch, i.e a C++ object is an instance of a class) and the use of the `poly~` Max MSP object which manages the polyphony and its own DSP consumption. With `poly~` several instances of the same patch can be defined and their processing activity can be totally taken off the DSP chain and CPU consumption with the *mute* message. Muting the different playing techniques algorithm decrease the CPU consumption significantly, however this solution is useful only in case of someone not using all the detections, but not in a complete use case. This functionality remains therefore useful but doesn't fit all cases. Another property of the `poly~` is that the DSP treatment can be specified to be done on all the processors of the comupter by using the *parallel 1* message.

However, despite all these options, the use of six abstractions instead of one `poly~` object with 6 voices remains faster. In his lightest version (no graphical object for output visualization), the detection algorithms used 6% of CPU for the DSP part against 9% with `poly~`. In a more friendly version (use of vumeter to monitor the signal and of other graphical elements to monitor the out-

[1]http://www.roland.com/products/en/GK-3/

puts of the detections), the consumption increases equally until, respectively, 10% and 13%.

In both cases, the main element which dropped down the CPU consumption was the decrease of the number of computed FFT. Indeed, before gathering all the detection, CPU consumption was around 50%, 60%. Depending on the situation the two solutions cited above can be used. If all the detections are used, the solution with the 6 abstractions fits best. In the case of not using all the detections, the solution using the `poly~` fits better.

Figure 8 shows the GUI of the patch used for the guitar playing techniques detections.

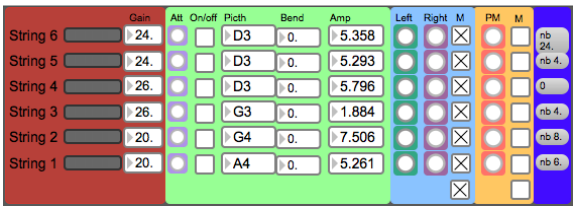


Figure 8: *Patch for guitar playing techniques detection (from left to right): input of each strings with adjustable gain, attack, pitch and note on / note off detection bend and amplitude tracking, left / right-hand attack discrimination, palm muted notes and harmonics detection.*

4.3. Mapping

It has to be noticed, before going into the details, that these mappings have been chosen in terms of the resulting sound they produced. Indeed, they were used in an improvisation framework where guitar and synthesized voice were mixed. Flexibility and sound quality were therefore what led the mappings designs. Moreover, at the time of the mappings' conception, no controls over the structure of the synthesized sentence or text was available, therefore the guitar could not have been used as a moving playhead or any similar type of control.

One of the main element used in the mapping between the guitar and the MAGE synthesizer is a 2D-interpolation tool (`node` Max MSP object). With this tool the states of the interpolation are defined by circles and distances are outputted as functions of the position of the cursor regarding each one of the states. The size of the circles is used as a weight applied to each states in the distances computation: the bigger the size, the bigger the state's influence. In [34], a similar tool has been developed and described. This interpolation tool can be linked to the `pattrstorage` object (store and recall presets) in order to easily use 2D-interpolation to move through defined presets. In our case, several voices with different parameters were defined as presets and the interpolation helped navigating between these voices, creating in-between voices when the cursor is in-between presets.

The second element that we added was a trajectory tool to record and play back movements (motion of the cursor in a lapse of time) into the 2D world of voices. Subjectively interesting (in terms of generated sound) trajectories were then recorded to be used as guitar-controlled materials. Figure 9 shows these 2D-interpolation and trajectory tools.

Three different mappings, controlling the trajectory tool, were defined. On each one of those mappings, volume of the guitar has been mapped to the volume of the synthesized voice, in order not to have sound unless the guitar is played. The three mappings are detailed below:

- first mapping: the note on the 3rd fret of the 4th string (F) loads a defined trajectory and the note on the 3rd fret of the 5th string (C) plays it. Any bends played on the 2nd and 3rd string is mapped to the vocal tract length; the bigger the string is bent, the smaller the vocal tract length is.
- second mapping: any normal attack on the 6th string chooses a trajectory and plays it. If time lapse between two palm muted notes is above a certain threshold the speed of the trajectory is 4 times faster.
- third mapping: an harmonic note on any of the strings overwrites the pitch of the voice. A series of four pitches are defined so that the voice follows a simple melody. Playing the note on the 2nd fret of the first string (F#) changes the series of pitches. As in the second mapping, the amount of bend is controlling the length of the vocal tract.

4.4. Discussion

The mappings used and listed above were a first attempt to give the MAGE synthesizer an intuitive controller.

On the guitar side, it appears that detection information needs to be reduced. Indeed, with this kind of setup (an instrument controlling synthesized or digital sounds) the player needs to keep a certain correlation between what he plays and what the audience sees and hears. In other terms, if a large amount of the detections is used and mapped to different elements of the synthesized voice, correlation between what is played on the guitar and what is heard can become blurry.

Moreover, being too specific about the mapping (i.e, a specific note played with a specific playing technique) can prevent the player from a certain flexibility and playability. In the improvisation context which was ours, these two elements were important to keep. In the third mapping e.g, using all the harmonic notes detection to overwrite the pitch was preferred to using only one specific harmonic note detection.

On the voice side, it appeared that changing the pitch of the voice was not that perceptible when the pitch difference was around a half tone to two tones. Mapping the

pitch of the guitar directly to the pitch of the voice, was therefore, not the best option.

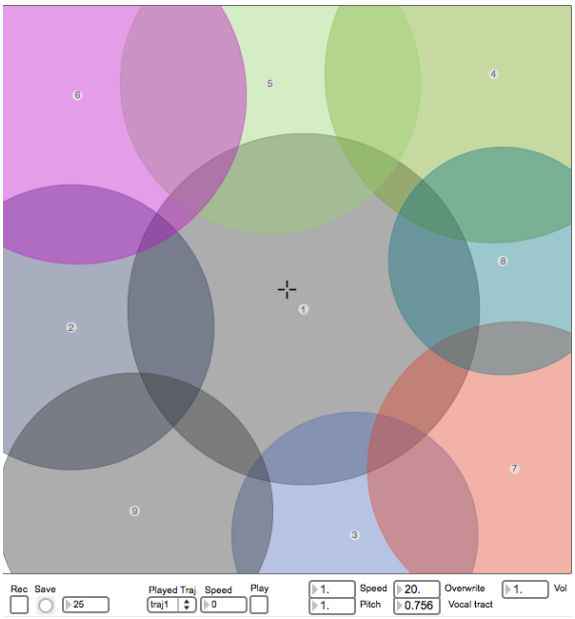


Figure 9: *2D-Interpolation and trajectory tools used to map the guitar detection to the* MAGE *synthesizer.*

5. Results

This project brought us to improve the two frameworks that we were using. On the one hand, we have been able to significantly improve both the computational load and robustness of the algorithms for extracting guitar playing techniques. Various useful expressive gestures such as hammers, pull-offs and harmonics can now be detected reliably on the six strings with a reasonable load on the computer. On the other hand, MAGE has been completely rewritten, leading to MAGE 2.0 being released soon, and this new version clearly leaps forward in term of synthesis reactivity. Due to the previous need in MAGE to preserve compatibility with the sentence-wise stream-based approach, many memory management issues were preventing MAGE to deliver a constant high-quality output with convincing reactivity. MAGE 2.0 makes everything far more usable for performative usages. As a consequence, real mapping strategies could be designed and tested during the workshop, such as using the tonal quality of the guitar to control the intonation of synthetic speech or mapping various fingering techniques to voice types and vocal effects.

6. Acknowledgements

Authors would like to thank the financial and academic supports of the MAGE project: University of Mons (grant 716631) and Acapela Group S.A. Alexis Moinet's work is supported by a public-private partnership between University of Mons and EVS Broadcast Equipment SA, Belgium. Also, we thank G. Wilfart for his contributions.

7. References

[1] B. C. J. Moore, L. K. Tyler, and W. D. Marslen-Wilson, Eds., *The Perception of Speech: From Sound to Meaning*. Oxford University Press, 2009.

[2] F. H. Guenther, S. S. Ghosh, and J. A. Tourville, "Neural Modeling and Imaging of the Cortical Interactions Underlying Syllable Production," *Brain and Language*, vol. 96, pp. 280–301, 2005.

[3] T. Dutoit, *An Introduction to Text-to-Speech Synthesis*. Klumer Academic Publishers, 1997.

[4] G. Fant, *The Acoustic Thenry of Speech Production*. Mouton The Hague, 1960.

[5] A. J. Hunt and A. W. Black, "Unit Selection in a Concatenative Speech Synthesis System Using a Large Speech Database," in *Proc. of the IEEE International Conference on Audio, Speech and Signal Processing*, 1996, pp. 373–376.

[6] B. Denby *et al.*, "Silent Speech Interfaces," *Speech Communication*, vol. 52, no. 4, pp. 270–287, 2010.

[7] O. Lähdeoja, "An approach to instrument augmentation : the electric guitar," in *Proceedings of the 2008 Conference on New Interfaces for Musical Expression (NIME08)*, 2008.

[8] L. Reboursière, C. Frisson, O. Lähdeoja, J. I. Mills, C. Picard, and T. Todoroff, "Multimodal guitar: A toolbox for augmented guitar performances," in *Proc. of NIME*, 2010.

[9] K. Tokuda, T. Kobayashi, and S. Imai, "Speech Parameter Generation from HMM Using Dynamic Features," in *Proc. of the IEEE International Conference on Audio, Speech and Signal Processing*, 1995, pp. 660–663.

[10] T. Yoshimura, K. Tokuda, T. Masuko, T. Kobayashi, and T. Kitamura, "Simultaneous Modeling of Spectrum, Pitch and Duration in HMM-based Speech Synthesis," *IEICE Transactions On Information And Systems*, vol. 83, no. 11, pp. 2347–2350, 1999.

[11] P. Cook, *Real Sound Synthesis for Interactive Applications*. AK Peters, 2002.

[12] I. Stylianou, "Harmonic plus noise models for speech, combined with statistical methods, for speech and speaker modification," Ph.D. dissertation, Ecole Nationale Supérieure des Télécommunications, 1996.

[13] T. Dutoit, *An Introduction to Text-To-Speech Synthesis*. Springer, 1997, vol. 3.

[14] A. Black, H. Zen, and K. Tokuda, "Statistical parametric speech synthesis," in *Proc. of the IEEE International Conference on Audio, Speech and Signal Processing*, vol. 4. IEEE, 2007, pp. 1229–1232.

[15] T. Fukada, K. Tokuda, T. Kobayashi, and S. Imai, "An Adaptive Algorithm for Mel-Cepstral Analysis of Speech," in *Proc. of the IEEE International Conference on Audio, Speech and Signal Processing ICASSP92 1992*, vol. 1, no. 1. IEEE, 1992, pp. 137–140.

[16] K. Tokuda, T. Masuko, N. Miyazaki, and T. Kobayashi, "Hidden Markov Models based on Multi-space Probability Distribution for Pitch Pattern Modeling," in *Proc. of the IEEE International Conference on Audio, Speech and Signal Processing ICASSP99*, vol. 1. IEEE, 1999, pp. 229–232.

[17] T. Yoshimura, K. Tokuda, T. Masuko, T. Kobayashi, and T. Kitamura, "Duration Modeling for HMM-based Speech Synthesis," in *Proc. of ICSLP*, vol. 2, 1998, pp. 29–32.

[18] H. Zen, K. Tokuda, and T. Kitamura, "Decision Tree based Simultaneous Clustering of Phonetic Contexts, Dimensions, and State Positions for Acoustic Modeling," in *Proc. of Eurospeech*, 2003, pp. 3189–3192.

[19] K. Tokuda, H. Zen, and A. W. Black, "An HMM-based Speech Synthesis System Applied to English," in *Proceedings of IEEE Workshop on Speech Synthesis 2002*, vol. 47, no. 1571. IEEE, 2002, pp. 227–230.

[20] K. Tokuda, T. Yoshimura, T. Masuko, T. Kobayashi, and T. Kitamura, "Speech Parameter Generation Algorithms for HMM-based Speech Synthesis," in *Proc. of the IEEE International Conference on Audio, Speech and Signal Processing*, vol. 3, no. 3. IEEE, 2000, pp. 1315–1318.

[21] S. Imai, K. Sumita, and C. Furuichi, "Mel log Spectrum Approximation (MLSA) Filter for Speech Synthesis," *Electronics and Communications in Japan (Part I: Communications)*, vol. 66, no. 2, pp. 10–18, 1983.

[22] J. L. Gauvain and C.-H. Lee, "Maximum A Posteriori Estimation for Multivariate Gaussian Mixture Observations of Markov Chains," *IEEE Transactions On Speech And Audio Processing*, vol. 2, no. 2, pp. 291–298, 1994.

[23] C. J. Leggetter and P. C. Woodland, "Maximum Likelihood Linear Regression for Speaker Adaptation of Continuous Density Hidden Markov Models," *Computer Speech and Language*, vol. 9, no. 2, pp. 171–185, 1995.

[24] T. Yoshimura, K. Tokuda, T. Masuko, T. Kobayashi, and T. Kitamura, "Speaker Interpolation in HMM-based Speech Synthesis System," in *Proc. of Eurospeech*, vol. 97, 1997, pp. 2523–2526.

[25] A. Sawabe, T. Yoshimura, K. Tokuda, T. Masuko, T. Kobayashi, and T. Kitamura, "Eigenvoices for HMM-based Speech Synthesis," in *Proc. of 7th International Conference on Spoken Language Processing ICSLP*, vol. 2, 2002, pp. 1269–1272.

[26] T. Nose, J. Yamagishi, T. Masuko, and T. Kobayashi, "A Style Control Technique for HMM-based Expressive Speech Synthesis," *IEICE - Transactions on Information and Systems*, vol. E90-D, no. 9, pp. 1406–1413, 2002.

[27] H. Zen, T. Toda, M. Nakamura, and T. Tokuda, "Details of the Nitech HMM-based Speech Synthesis System for the Blizzard Challenge 2005," *IEICE Transactions on Information and Systems*, vol. 90, no. 1, pp. 325–333, 2007.

[28] H. Zen, K. Tokuda, and A. W. Black, "Statistical Parametric Speech Synthesis," *Speech Communication*, vol. 51, pp. 1039–1064, 2009.

[29] F. Iazzetta, "Meaning in music gesture," *Trends in gestural Control of Music*, 2010.

[30] R. Graham, "A live performance system in pure data: Pitch contour as figurative gesture," in *Proc.of Pure Data Convention*, Bauhaus-Universität, Weimar, Germany, August 2011.

[31] E. Guaus, T. Ozaslan, E. Palacios, and J. L. Arcos, "A left hand gesture caption system for guitar based on capacitive sensors," in *Proc. of NIME*, 2010.

[32] C. Traube and P. Depalle, "Extraction of the excitation point location on a string using weighted least-square estimation of comb filter delay," in *Proceedings of the Conference on Digital Audio Effects (DAFx)*, 2003. [Online]. Available: http://www.elec.qmul.ac.uk/dafx03/proceedings/pdfs/dafx54.pdf

[33] L. Reboursière, O. Lähdeoja, T. Drugman, S. Dupont, C. Picard, and N. Riche, "Left and right-hand guitar playing techniques detection," in *Proc. of NIME*, 2012.

[34] T. Todoroff and L. Reboursière, "1-d, 2-d and 3-d interpolation tools for max/msp/jitter," in *Proc. of NIME*, 2009.

CITYGATE
The multimodal cooperative intercity Window

Radhwan Ben Madhkour, Pierluigi Dalla Rosa, Marek Hruz, Miroslav Jirik, Ambroise Moreau, Ivan Pirner, Tomas Ryba, Jakub Vit, Francois Zajga, Petr Zimmermann.

***Abstract*—CityGate aims at developping a tool for enhancing telepresence between cities. We based our system on Scenic, a telepresence software developped in Linux. Our goal is to add games, interactions design and artistic installations possiblities. To achive these goals, we developped a system for streaming custom images instead of images captured by a camera, such as videos, generated images, preprocessed camera images and so on. Using this system, we developped a first collaborative game. It is a Pong-like , controlled by the position of the face. On an other side, we also started to develop a system for the remote control of the streaming. At the end, this software will allow to launch/stop streaming, enable/disable display, etc from a remote computer and also to ease configuration of the system via a graphical programming interface (inputs, outputs and processing blocks).**

***Index Terms*—Telepresence, Cities, Streaming, Art**

I. Introduction

MONS and Plzen will be the European capitals of culture in 2015. Their respective mainline will be When Technology Meets Culture and Pilsen, Open Up!.

In order to prepare this event, UMONS and UWB have started collaborating on digital art technology, starting with the KINACT [1] project during eNTERFACE11 in Plzen.

One of the activities that could be organized between cities as part of the 2015 event would rely on establishing creative interaction between citizens in both cities. This project implies to build a common infrastructure for allowing real-time multimodal interaction. The main goal of the CITYGATE project is to achieve a first step in this direction, by developing the technology components required for interaction.

More precisely, the CITYGATE project will allow:

- Audiovisual telepresence streaming,
- Interaction: Games, Dance and Music performances, VJing / DJing ...,
- Cooperative multiplayer (social) games (like in KINACT),
- Digital art installation.

The rest of the paper is organised as follows. Section II describes the Citygate architechture. Section III presents the Scenic telepresence softawre [2], an open source software on which Citygate is based. Section IV introduces the use of video devices in linux and especially explains the loopbackvideo for linux devices. Section V is dedicated to the description of the shared memory, an option of Scenic for sharing the image sent with other software. Section VI gives details about our first tests. Section VII describes the first collaborative games implemented using the telepresence software. Section VIII shows how we have thought the remote control interface. Finally, Section IX concludes the work and gives some perspectives to improve the system.

Radhwan Ben Madhkour, Ambroise Moreau and Francois Zajga are with the TCTS lab, University of Mons, Mons, Belgium.

Marek Hruz, Miroslav Jirik, Ivan Pirner, Tomas Ryba, Jakub Vit and Petr Zimmermann are with the department of cybernetics, University of West Bohemia, Plzen, Czech Republic.

Pierluigi Dalla Rosa is with the Istituto Superiore Mario Boella, Torino, Italy

II. Software Architecture

The citygate software will be decomposed in 4 parts:

Core The core of the Citygate is the streaming system. This system is based on the software Scenic. The goal is to be able to stream custom images and sound. The image could be simply a webcam streaming, a synthesized image or a processed camera image. In the case of the sound, it's already possible to send custom sound using PureData or other software using a jackd server.

Connections In order to let artits take the full control of the system and send their real time performances for their favorite framework, connections with common librairies and frameworks (OpenFrameworks [3], Processing, PureData, etc) and Citygate are required. Image and audio streaming from those apps to remote client through Citygate will be implmented.

Remote control The remote control is a set of commands for controlling the system remotly. These commands include a port and ip settings, camera selection, streaming launch action, display switch on/off, ...

Plugins Games, permanent installations and other softwares would be installed as plugins in the system.

Figure 2 shows the normal use of Scenic. The goal of this project is to achieve a system like the one presented in figure 1 with multiple connections with third party "creation" softawre, a plug-in abillities to allows games and other process and a remote control interface.

III. Scenic

Scenic is an opensource telepresence software developed by the SAT (Society for Arts and Technology, Montreal). It allows to do visio conference with multiple audio canals and additional MIDI control. Scenic is composed by two parts: a GUI written in Python and a streaming program called "milhouse" written in C++.

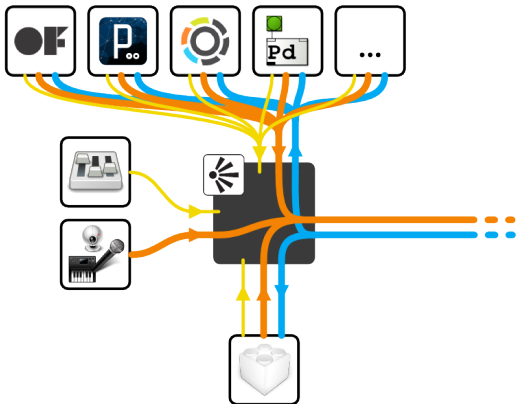


Fig. 1: Citygate Architecture

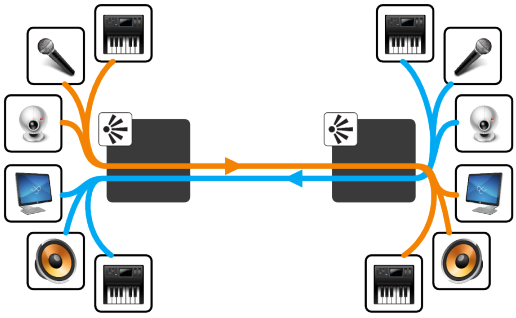


Fig. 2: Scenic architecture. Users can send/receive image, sound and midi control. Images are acquired via v4l camera, DV camera or firewire. The acquisition of the sound is done with a mix table through a jackd server.

Scenic is optimized for the collaborative artistic performance. The synchronization between audio and video is accurate and the streaming system is efficient.

IV. Loopback Device

In linux, each connected video device has a video buffer. This video buffer is created in the /dev directory when a device is connected. The buffer is named /dev/videoX in the case of a video4linux device, X being the number of the device (eg. /dev/video0 for the first device connected, /dev/video1 for the second device connected, /dev/videoN for the n^{th} device connected).

In our case, we want to send custom images. Hence, Scenic has to connect to a virtual video device. This device is created with video4linux loopback device. A loopback device is a video buffer created by the user and not the kernel. Even if there is no camera connected, a loopback device can be created and Scenic considers it as a valid video devices. Custom images are created and pushed in the loopback buffer. For a correct transmission, the image has to be adapted to the device color space and subsampling model (e.g. YUV422 or YUV420 depending on the buffer settings).

V. Shared Memeory

With the loopback devices, we can send custom images. Nevertheless, in case we want to process the received image before it is displayed, an access to the received image is needed. Scenic has the hability to push the image received in a video buffer. This buffer can be read using the library libshmdata.

VI. Applications and tests

We tested different configurations.

1) From OpenCV to Scenic
2) From Scenic to OpenCV
3) From OpenCV to OpenFrameworks
4) From OpenFrameworks to Opencv

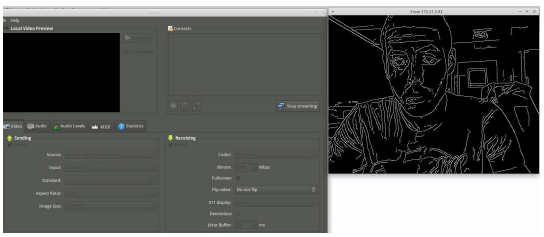
Figure 3 shows the results for different scenarios tested. In the case of OpenCV [4] transmission (figure 3a), we work on a Mat or IplImage. Once all the processing is done, we push the data using the pointer on the array. For OpenFrameworks, the rendered image or region corresponding to the loopback device resolution is grabbed and pushed in the buffer.

VII. Plugins

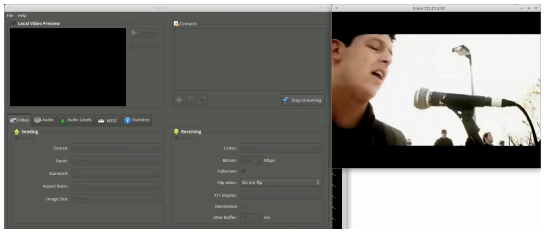
The pong game was chosen as an example of implementation of the plug-in system. It uses the library OpenCV. The rules of the pong game are simple. Two players play against each other each player. They are located on one side of the game window. The player controls a pad's vertical location. The game begins with a stationary ball in the middle of the playing field. The ball then moves in a random direction towards one of the players. The goal of the game is to hit the ball with the pad and send it to the other side so that the other is not able to play back. If the ball passes one players side the other player gets a point. The pad is controlled by the movements of the player's head. The code of the game is utilizing the plug-in system developed during the workshop. The main class GamePingPong is inherited from the class GameLogic. The GameLogic class is a virtual class with one method doStep. In current implementation, the class takes two images into account; one local image captured on a local machine and one remote image send via net. The implementation can be easily rewritten to take 1..N images into account. The doStep method should define what happens in one frame of the game. In case of the Pong game, one step of the game is composed of the call of two method: prepareImageAndDetectFace and ping_pong. Both method are explained bellow.

A. Method prepareImageAndDetectFace

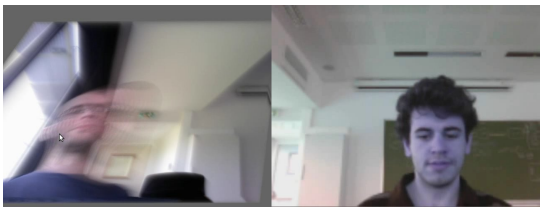
This method converts the original RGB image into a gray-scale image and resizes it. For now we use a scale factor 0.5 which makes the image of quarter area. We equalize the histogram of the image for a better face detection accuracy. To detect the face we use our FaceDetector class which utilizes the OpenCV function detectMultiScale. When the face is detected its coordinates are transformed back to the original coordinate system of the image a returned as an output of the method. If the face is not detected the face structure entries are set to zero. The method is called twice, once for local and once for remote images.



(a) Opencv video grabber (v4l based), edge detection and transmission.



(b) Simple video grab and transmission



(c) OpenFrameworks app. On the left, Openframeworks local grab (frame are bufferized in a FILO and displayed with a transparency). On the right, the frame transmitted via Scenic is acquired in OpenFrameworks through the shared memory and displayed.

(d) The OpenFrameworks left part of the screen in figure (c) is transmitted. This figure show the difference between the local image grabbed and the image received from OpenFrameworks. The code is written with OpenCV.

Fig. 3: First results and tests

B. Method ping_pong

This method is the core of the game logic. It is provided with both images and face regions. In the first step (or frame) the game is initialized. The necessary images are loaded from hard disk. It is the background image (backgrounf.png) and an image of the ball (ball.png). The size of the playing field is given according to the size of the background image. The collision detection rectangle size of the ball is given according to the size of the ball image. The images of pads are also loaded from the disk (pad.png) and the collision rectangles are set for players according this image. This concludes the initialization step which is not repeated during the game. Next, the game analyzes the output of the face detection algorithm. If there was no face detected for several frames (in current implementation 10) the AI takes over. The AI is perfect in the sense that the pad is always following the ball. It is limited only by the maximal allowed speed of the pad. If a face was detected the center of the player's face is the desired destination of the pad. We calculate whether the transition from current location to the desired one is allowed with respect to the maximally allowed speed. If not the transition is truncated to the maximally allowed value. This creates a certain lag in the movement which makes the game more difficult to master. This process is repeated for the other player too. Next, the ball is moved. The collisions are computed using the collision detection rectangle. If the rectangle of the ball passes the edge of the screen the player on the opposing side is given a point. If the rectangles of the ball and the pad collide the new direction of the ball is computed. The x component of the speed vector of the ball is negated so that the ball moves in the opposing direction. The y component is chosen randomly to give a little twist to the game. After this analysis the result is rendered to the screen and the process is repeated from point 1.

C. Discussions

The first tests are conclusive and the game is fun to play even if it is simple. Nevertheless, if the seond player is far from the second one, a delay could appear and affect the gameplay. To solve this issue, we intend to implement a server client architecture where each player is a client and the server is managing the game. Moreover, a true plugin architecture has to be put in place to replace the current "function-based" plug-ins. Figure 4 shows the final results of the Pong game

VIII. Remote control

The remote controller is an important element in the whole infrastructure because it is the main interface of the core system. The interface should show actions for pre-processor and enable the input to be activated.The main inspiration for the functional positioning of element comes from DJ consoles and from node base editor.

There is a list of inputs on the left, every input could be connected or disconnected from the outputs that are shown as an other list on the right. The implementation is done in OpenFrameworks.

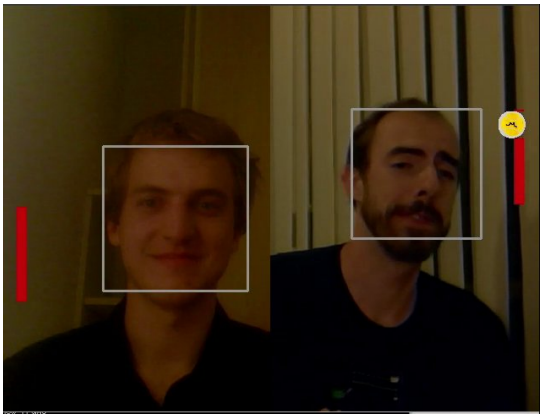


Fig. 4: Network Pong game controlled with the face position

The operations that could be done between inputs and outputs are:

- connect an input to an output;
- delete the connection;
- set a value on a preprocessor.

The communication with the core system is made thanks to Open Sound Control (OSC).

During the initialization the remote control ask for the list of inputs and visualize them. The same happens for the list of outputs that is provided by the core system. In the prototyping phase was build a simulator for the core system that gave to the remote controller the information about inputs and outputs at startup.

Every time a module is added or a connection is done the remote controller send a message to the core system to notify the change.

Moreover it is possible to visualize in the interface of the remote controller the frame rate of the application, or it is possible to set some action like start/stop or any other action could be defined. The list of inputs notify the type with an icon that explicitly help understand which kind of device is behind. The integration of the remote controller with the preprocessor depends strictly with the input device, it is meant to be linked with OSC.

The main idea is to build a node editor that could allow the maximum modularity and expandability of future development.

IX. Conclusion

During this project, we developped a system for streaming custom images and audio. This system is based on Scenic, a telepresence software implemented by the SAT, Montreal (CA). Our system extends Scenic with the possibility of connecting it with different tools. A Pong-like game was developped to demonstrate the possibility of interaction. We also started to create a remote control to change settings and manage the system from a remote place.

Acknowledgment

The authors would like to Supelec Metz and especially Malis team for organizing the eNTERFACE summer Workshop.

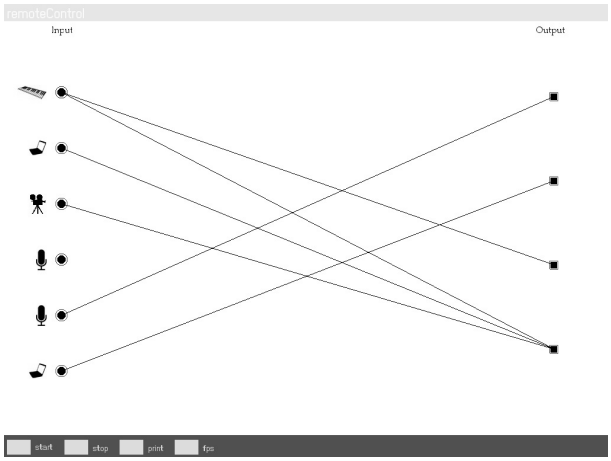


Fig. 5: Interface to connect input and outputs

References

[1] M. Mancas, R. B. Madhkour, D. D. Beul, J. Leroy, N. Riche, Y. Rybarczyk, and F. Zajga, "Kinact: a saliency-based social game," in *Proceedings of the 7th International Summer Workshop on Multimodal Interfaces eNTERFACE11*, 8 2011.

[2] (2012) Secnic. [Online]. Available: http://code.sat.qc.ca/redmine/projects/scenic/wiki

[3] (2012) Openframeworks. [Online]. Available: http://www.openframeworks.cc

[4] (2012) Opencv: Open source computer vision library. [Online]. Available: http://opencv.willowgarage.com/wiki/

P8 - Active Speech Modifications

Yannis Stylianou, *Valerie Hazan*, Vincent Aubanel, Elizabeth Godoy, Sonia Granlund, Mark Huckvale, Emma Jokinen, Maria Koutsogiannaki, Pejman Mowlaee, Mauro Nicolao, Tuomo Raitio, Anna Sfakianaki, Yan Tang

1 Introduction

In many intelligibility studies, it was demonstrated that the speaking style referred to as clear speech is significantly more intelligible than conversational (or casual) speech. This intelligibility gain exists for both normal-hearing and hearing-impaired listeners (e.g. elderly persons and linguistically inexperienced listeners like non-native (L2) speakers and children). Also, in a two-way conversation in which one person is affected by an adverse listening condition and one is not (e.g. between one person speaking to another via telephone where the other is in a noisy club, or in a cafeteria, in the street etc.), the person who is not affected still manages to make adaptations (on acoustic-phonetic and linguistic levels) that are quite specifically tailored to counteract the specific communication barrier that the other person is experiencing. These adaptations show that clear speech is not defined in a uniform way, but that there are different styles of clear speech depending on the adverse condition that the speech is heard in. In this context, Active Speech Modifications refer to the speaking-style adaptations or strategies a speaker applies in order to maximize communication effectiveness.

Identification and effective manipulations of the most prominent acoustic-phonetic characteristics of different styles of clear speech can allow for the development of new, signal based, active speech modification algorithms to increase intelligibility. The algorithms can consequently improve speech intelligibility in many situations, such as in the design of hearing aids, telephony, and other speech signal processing technologies and applications (i.e., speech synthesis, recognition, enhancement, etc).

The purpose of this project was to use modern speech analysis and reconstruction algorithms to:

- identify which acoustic-phonetic characteristics are prominent in different styles of clear speech (e.g. babble-countering clear speech, vocoder-countering clear speech, L2-"countering" clear speech) and when they are realized in time.
- model a selection of these aspects so that they can be applied automatically on speech, to enact prosodic changes, changes in amplitude spectrum, modulation frequencies, etc..
- run a series of "proof of concept" perception experiments to see if the "specifically-enhanced" speech is better perceived in the "matched" adverse condition than other types of clear speech (there is evidence that this is the case with the naturally-enhanced speech).

The outcome of the project can be summarized as follows:

- a new speech corpus (P8-Harvard corpus) was linguistically and meta-linguistically annotated and acoustically analyzed with the goal of identifying which acoustic-phonetic characteristics differ between clear and casual speech and also between different styles of clear speech. Moreover, acoustic analyses on specific features were also performed on a different corpus, namely the LUCID database (specifically on read clear and read casual speech signals).
- among the different styles of clear speech, prosodic changes were most apparent. Therefore, signal modification algorithms were developed to mimic human adaptations on prosody in adverse conditions with the aim of increasing intelligibility.
- a user-friendly interface, XPlic8, for a large range of acoustic analyses was developed.
- a set of evaluation experiments was prepared to evaluate the different modifications.

This report is organized as follows. Section 2 describes the P8-Harvard corpus that contains the different speaking styles for analysis. In section 3 the linguistic analysis of the corpus is presented. Section 4 focuses on the analysis of the voice source characteristics between different styles of speech on the P8-Harvard corpus (and on the LUCID corpus to a less extent). In section 5 prosodic differences between the different speaking styles are examined with focus on the number of pauses and the mean word duration. Section 6 introduces two novel time-scaling techniques that try to modify casual speech signals to achieve higher intelligibility scores, mimicking the properties of the elicited clear speech. Section 7 presents a novel tool for a large range of acoustic analyses. Section 8 summarizes the work of this project.

2 P8-Harvard Corpus design and recording

A corpus of materials was recorded and analyzed to provide information about the acoustic phonetic enhancements typically seen in clear speaking styles produced in speech with communicative intent. The aim was to record materials which were controlled and standardized (Harvard sentence lists) but where clear speaking styles were elicited naturally, due to communicative need, rather than via instructions to read materials clearly (LUCID corpus[1]). For that purpose, the first 15 lists of the Harvard sentences (1969) were recorded. These sentences, which are phonetically-balanced and each include 5 keywords, were developed for speech quality evaluations.

2.1 Recording procedure

Two British English speakers, one female and one male were each recorded (as "Speaker A") with a confederate ("Speaker B"). Speaker A had to read a sentence to Speaker B who had to repeat it back to Speaker A. So as to induce Speaker A to make an effort to speak clearly when Speaker B was experiencing a communication barrier, speakers were told that the speaker pair that achieved best "intelligibility scores" would win a prize. Speaker A was told to only say the sentence once even if errors were made by speaker B in repeating it. Two types of communication barrier, following Hazan and Baker (2011), were used in order to elicit clear speaking styles that may differ somewhat in their acoustic-phonetic characteristics. In the "babble" (BAB) condition, Speaker B heard speaker A's voice mixed with 8-speaker babble noise at an approximate level of 0 dB SNR; in the "vocoder" (VOC) condition, Speaker B heard speaker A's voice passed through a three-channel noise-excited vocoder which spectrally degraded the signal. 150 sentences in each of the three conditions ("no barrier" NB, BAB, VOC) were recorded for the two speakers.

Speakers were seated in separate sound-treated rooms. Beyerdynamic DT297PV headsets fitted with a condenser cardioid microphone were used and the speech was recorded on two separate channels at a sampling rate of 44100 Hz (16 bit) using an EMU 0404 USB audio interface and Adobe AUDITION. Only Speaker A's output was analysed here, since speaker A was talking in a non-barrier environment.

3 Linguistic analysis of the P8-Harvard Corpus

For the linguistic analysis of the P8-Harvard corpus, Praat [2] along with several analysis algorithms was used.

3.1 Initial processing

For all sentences, a Praat textgrid was produced with three tiers: tier 1 contains speech (SP) and silent (SILP) regions markers, tier 2 had word aligned markers and tier 3 phoneme-level aligned markers. Sentences (five sentences for Speaker A1 and 12 for Speaker A2) were excluded from the corpus since they contained mispronunciations or hesitations on one or more of the keywords.

3.2 Linguistic Annotation of corpus

The Harvard database [3] is a set of 72 phonetically balanced lists of 10 sentences, each containing 5 keywords. Three lists were recorded for the current project, and in addition to existing keyword coding, the database was enriched with broad/narrow grammatical annotation, lexical frequency and neighborhood density. A summary of the added information to the Harvard database is given in Table 1. Word- and phone-level annotation were semi-automatically carried out and merged with the Harvard

database. The resulting corpus comprises of 2293 manually check words and 6902 segments for the two speakers in the three recording conditions.

Information	Description
word	Orthographic form of the word (punctuation removed)
lemma	Lemma of the word
keyword	Keyword coding of the word (keyword vs. non-keyword)
PoS	Part of speech. Categories are: Adj, Adv, Conj, Det, DetP, Ex, NoC, Num, Prep, Pron, Verb, VMod
freqBNC	BNC[1] frequency of occurrence of the word (inflected form). Occurrence per million in a 100 million spoken and written word corpus
neighPhon	Number of all phonological neighbours that differ from the word by a 1-phoneme substitution, deletion, or addition. Extracted from the de-Cara database[2].
freqCxS	Celex spoken frequency of corresponding lemma. Occurrence per million in a 17.9 million spoken word corpus
freqCxW	Celex written frequency of corresponding lemma. Occurrence per million in a 17.9 million written word corpus

Table 1: Harvard database annotation tagging. 3 lists were annotated for a total of 1066 words.

3.3 Analysis of communication effectiveness

The number of correctly-transmitted keywords was calculated per condition. The percentage of keywords correct in the BAB condition was 88% for A1 and 73 % for A2, while in the VOC condition it was about 40% for both speakers. The VOC condition was therefore harder for both speaker pairs.

Communication breakdowns were defined as sentences in which 3 or more keywords were not correctly repeated by speaker B. Fig. 1 highlights these breakdowns and the sentences immediately following them (see also Section 4.6).

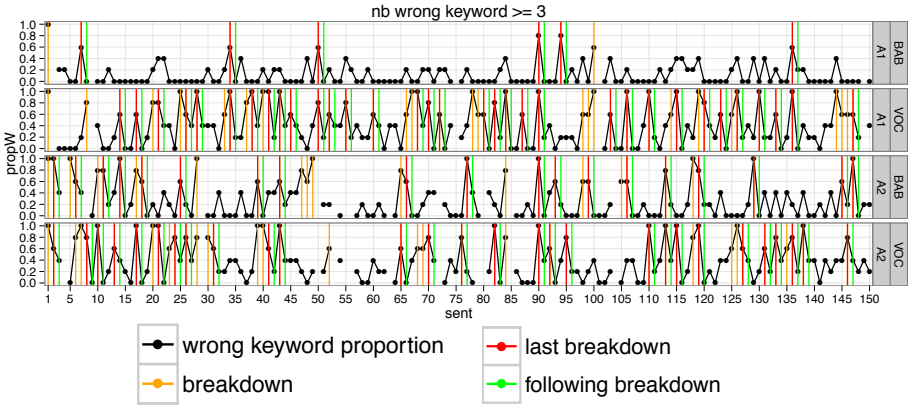


Figure 1: *Identification of communication breakdowns for speaker A1 and A2 in BAB and VOC conditions, defined as sentences in which more than 3 keywords were missed in the interlocutor repetition. A distinction is made between ''breakdowns" (orange lines) and ''last breakdowns" (red lines), the latter depicting breakdowns immediately followed by sentences in which 2 or less keywords were missed (''following breakdowns", green lines).*

[1]British National Corpus. Available online at `http://ucrel.lancs.ac.uk/bncfreq`
[2]de-Cara database. Available online at `http://portail.unice.fr/jahia/page12414.html`

4 Analysis of the voice source and spectral characteristics between different styles of speech

The analysis of voice source characteristics of the P8 corpus included three types of speech: NB, BAB and VOC speech. The idea was, that if there would exist differences between the voice source characteristics of these two voice types, this information could used to convert normal speech into the more intelligible speech in the barrier cases.

The main analysis tools for this task were glottal inverse filtering, pitch detection, glottal closure instant detection, voice source feature extraction and formant detection using Praat. These tools were developed as Matlab scripts for the purpose of the project and details regarding the analysis tools are provided in section 7, since all these analysis algorithms were incorporated in a new proposed analysis tool.

4.1 Glottal flow waveforms and Harmonic analysis

Main findings were that the different voice types did not differ significantly in terms of the use of the voice source. Figure 2 shows the glottal source waveform for the speech signals on the three different conditions, BAB, VOC and NB for the male and female speaker. In Figure 3 the corresponding spectra of the glottal source are depicted. Figure 4 shows a slight decrease of the harmonic-to-noise ratio in the barrier cases for both speakers A1 and A2.

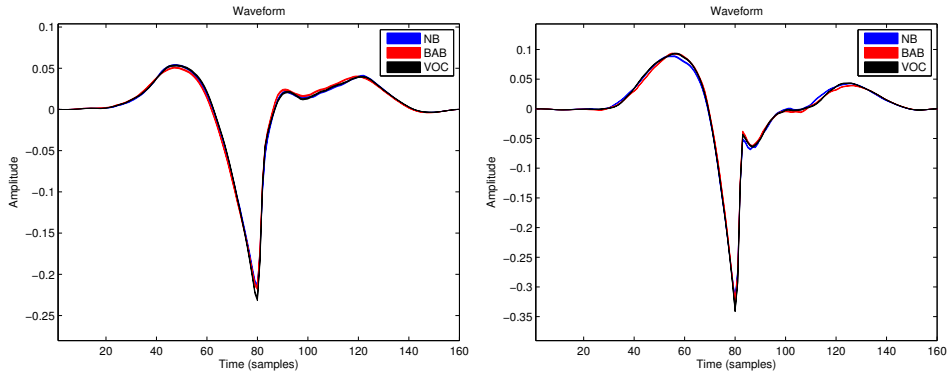


Figure 2: *Glottal source waveform for the speech signals on the three different conditions, BAB, VOC and NB for the male(a) and female speaker(b)*

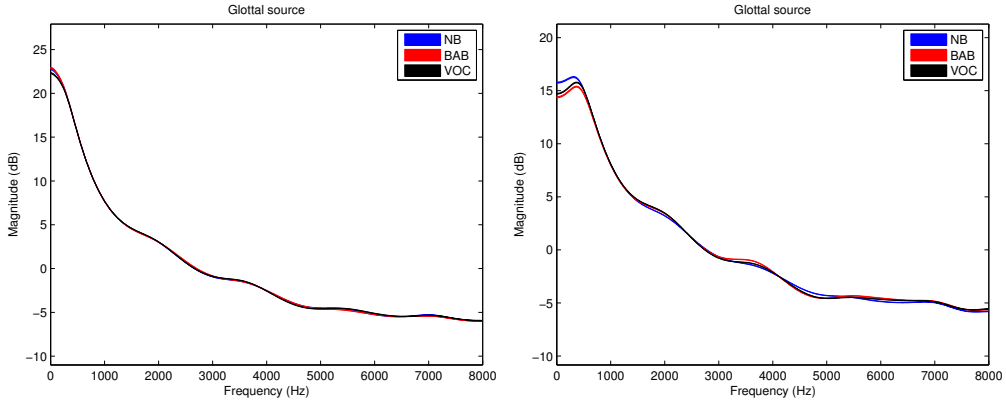


Figure 3: *Glottal source waveform for the speech signals on the three different conditions, BAB, VOC and NB for the male(a) and female speaker(b)*

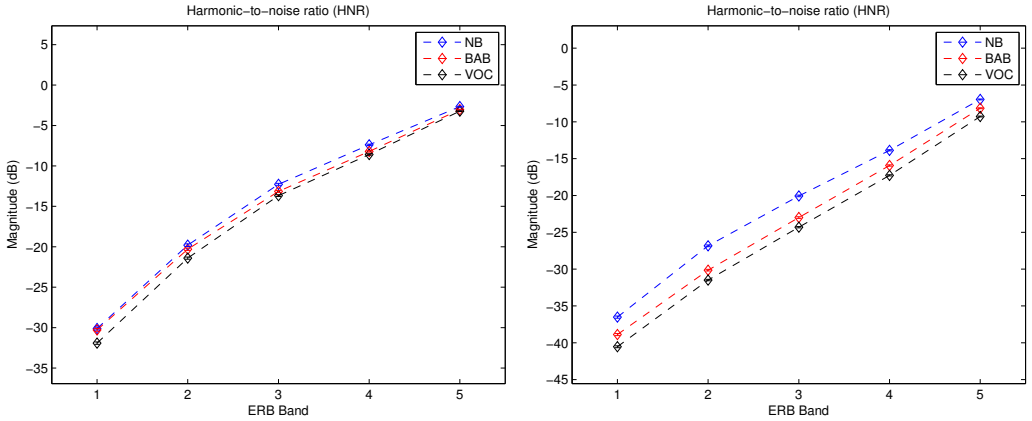


Figure 4: *Harmonic-to-noise ratio for the three different conditions NB, BAB, VOC for the male speaker A2 (left) and the female speaker A1 (right).*

4.2 F0 analysis

The F0 detection is based on glottal inverse filtering and autocorrelation peak detection. The algorithm implemented to extract the F0 and the F0 range from the speech signals is described on section7. These estimated values for the the whole P8-Harvard corpus where statistically analyzed with ANOVA. As expected F0 median was higher for the female speaker $[F(1,137) = 16343.6, p < 0.001]$; it was also higher in the VOC condition than in the NB $[t = 19.3; p < 0.001; df = 132]$ and BAB conditions$[t = -10.7; p < 0.001; df = 132]$, and higher in the BAB than NB conditions $[t = -7.6; p < 0.001; df = 132]$. F0 range also varied across conditions $[F(2,274) = 9.5; p < 0.001]$: it was broader in BAB than in both NB $[t = -2.4; p = 0.018; df = 132]$ and VOC $[t = 4.8; p < 0.001; df = 132]$. However, F0 range did not differ between the NB and VOC conditions $[t = 1.8; p = 0.067; n.s.; df = 132]$. The interaction between speaker and condition was also significant $[F(2,274) = 3.9; p = 0.02]$.

4.3 LTAS

The Long Term Amplitude Spectra (LTAS) were also estimated for the P8-Harvard and the LUCID corpus (the algorithm for the estimation of LTAS is described in Section 7. Previous studies correlate the increase of intelligibility of clear speech with the higher energy in the frequency band 1-3kHz relative to casual speech. Figure 5 depicts the LTAS for speakers A2 (left) and A1 (right) correspondingly for the barrier and no barrier conditions of the P8-Harvard corpus. The male speaker increases his energy above 1000Hz especially for the VOC condition and less on the BAB. For the female speaker there is a slight increase between 2000-4000Hz for the BAB condition and a significant increase above 5000Hz.

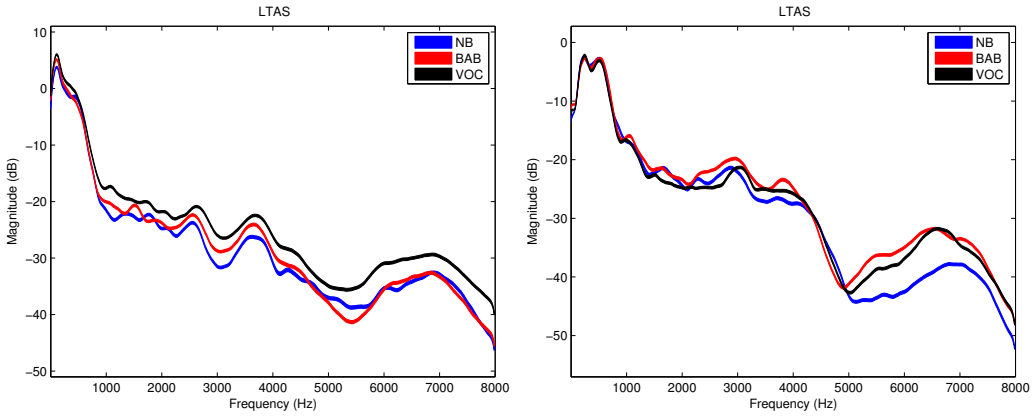


Figure 5: *LTAS of the male (left) and female speaker (right) for three different conditions NB, BAB and VOC*

For the 21 speakers in the LUCID database, averages over all voiced frames of the speaker were computed. The obtained results indicated that for most speakers, the spectral tilt decreases from CV to CL speech. In addition, some energy reallocation to the 1-7 kHz frequency region took place for most speakers. An example of a computed LTAS is shown in Fig. 6 for speaker F38 where the previously mentioned effects can clearly be seen.

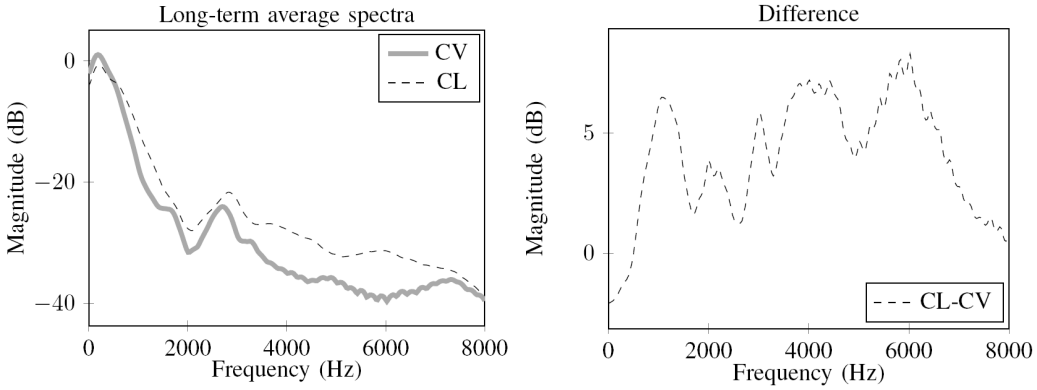


Figure 6: *The long-term average spectra (LTAS) of conversational (CV) and clear (CL) speech and their difference for female speaker F38 in the LUCID database. The LTAS are computed over four sentences for each condition.*

However, the results also varied significantly across speakers. For instance, the energy reallocation patterns were in most cases very different and furthermore, for some speakers the spectral tilt was further increased. This indicates that the speakers used very different strategies to produce clear speech.

A repeated measures ANOVA was done on the measure of intensity (LTAS $1-3kHz$) for the P8-Harvard corpus. LTAS was calculated separately for each sentence. There was a main effect of speaker $[F(1,131) = 22.0; p < 0.001]$, and of condition $[F(2,262) = 547.8; p < 0.001]$; post-hoc paired t-tests show that the BAB condition was greater in intensity ($mean = -3.1$) than the VOC (mean= -3.6) ($t = 4.8; df = 131; p < 0.001$) and NB conditions ($mean = -6.9$) ($t = -26.8; df = 131; p < 0.001$). There was also a significant interaction of speaker and condition $[(F(2,262) = 124.7; p < 0.001]$; post-hoc analyses show that there are significant speaker-specific strategies in terms of intensity ($t = -15.4; df = 131, p < 0.001$): for A1, the BAB condition has a greater intensity than the VOC condition (mean difference between VOC and $BAB = -2.3$), while for A2, the VOC condition has a greater intensity than the BAB condition (mean difference between VOC and $BAB = 1.2$).

4.4 Energy distribution in critical bands

A sinusoidal signal analysis/synthesis mode was used to check the differences between the clear and causal speech on the LUCID corpus. The idea was to investigate the differences between the two speaking styles, clear and casual speech, according to their sinusoidal features (including amplitude and frequency) extracted at the designed critical bands. For this, a pitch-independent sinusoidal model is designed which extracts one sinusoid per critical bands, hence with a fixed dimension equal to the number of critical bands. To design the critical frequency bands we used the 24 center frequencies and bandwidth derived at 16 kHz of sampling frequency. In order to reflect more accurately the subjective loudness of speech signal for the masker noise, the $ITU - R468$ noise weighting filter was taken into consideration. The highest spectral amplitude per frequency band was selected to avoid sidelobe peak problem. This modified the center frequency and bandwidth of some of the critical bands. The sinusoidal model designed as such showed a hardly distinguishable difference between the re-synthesized and the original signal.

Experiments were conducted on voiced frames of length 16 ms with a frame shift of 4 ms for two speakers, M8 (male) and F22 (female), of the LUCID database. Figure 7 shows the histogram of the amplitude (top) and frequency (bottom) of clear (blue) and casual speech (red) at a specific critical sub-band characterized with its center frequency and bandwidth for the male speaker M8. Figure 8 shows the histogram of the the amplitude (top) and frequency (bottom) of clear (blue) and casual speech (red) at a specific critical sub-band characterized with its center frequency and bandwidth for the female speaker F22. The center frequency and bandwidth for each critical band is shown at top of each subplot.

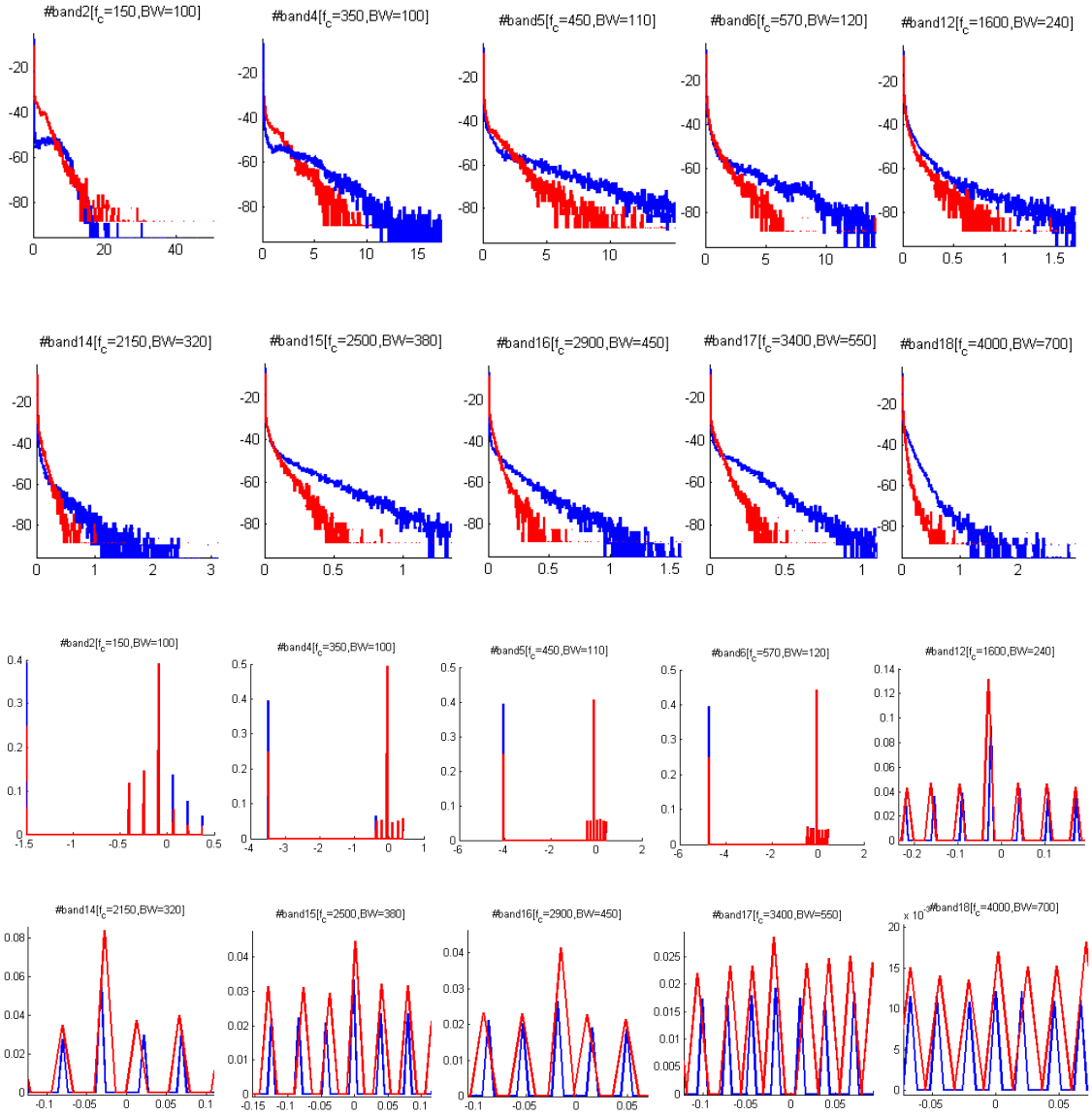


Figure 7: *Histograms of the amplitude (top) and frequency (bottom) of clear (blue) and casual speech (red) at a specific critical sub-band characterized with its center frequency and bandwidth for the male speaker M8*

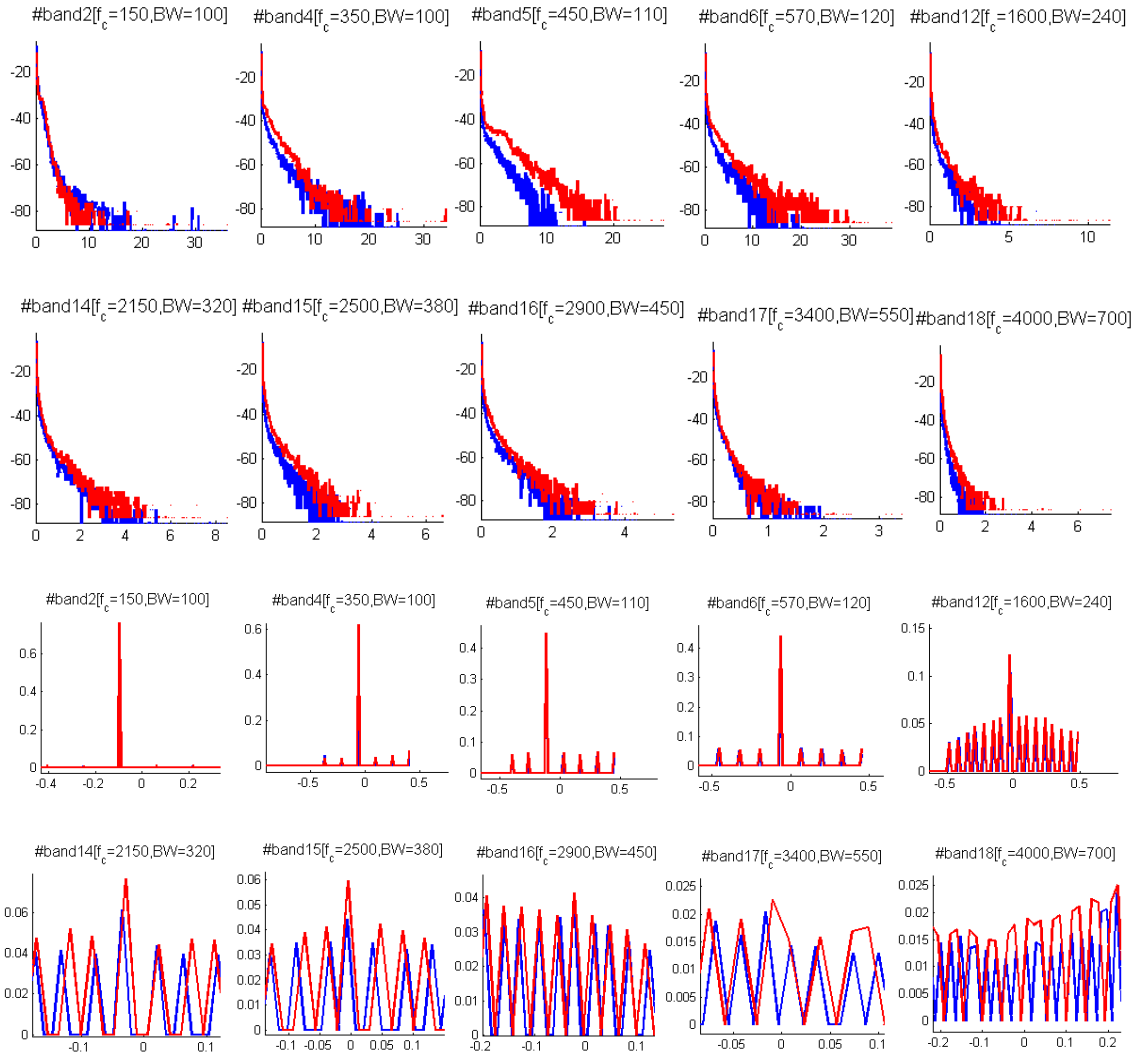


Figure 8: *Histograms of the amplitude (top) and frequency (bottom) of clear (blue) and casual speech (red) at a specific critical sub-band characterized with its center frequency and bandwidth for the female speaker F22*

The x-axis of each subplot is the range for amplitude or frequency. For frequency case, the x-axis is the frequency deviation from the center frequency (f_c) normalized by the bandwidth (BW) to make it a standard random variable called ($f_{standard}$):

$$f_{standard} = (f - f_c)/BW \tag{1}$$

The amplitude histogram figures, indicate the amount of energy difference per critical band between clear speech and casual speech. It is observed that for clear speech we have significantly more energy contribution than that for casual speech. This is well pronounced for frequency bands lying higher than 450 Hz. Looking at the changes in the frequency of clear and casual speech at critical bands, it is observed that the two speech styles have differences at frequency bands between 2000 and 4000 Hz (critical bands 13 to 18).

Future analysis can be performed in this domain. The idea is to find a way to model these differences between the barrier and no-barrier speech. Using the learned statistics, the final goal is to modify the barrier speech (causal speech), in terms of its sinusoidal parameters at critical bands, in order to improve the speech intelligibility. One possible idea is to increase the energy distribution of the causal speech at certain critical bands.

4.5 Vowel space

In order to visualize and quantize the vowel pronunciation of different speakers and styles, vowel spaces are useful. The vowel space is a plot of the mean of vowel instances in a 2D plane defined by the first and second formant frequencies. The area that the observed vowels span in this space then reflects the discriminability of the vowels. Previous studies report the expansion of vowel space in the case of clear elicited speech versus casual speech. The vowel spaces have been generated as follows. First, in order to isolate the vowel instances in the corpora, all of the speech was segmented using an HTK-based audio-to-text aligner. No manual corrections were performed. For each vowel instance, formant analysis is performed using the Praat algorithm [2]. The representative pair of F1 and F2 values for each vowel instance is then taken as the values at the center of the speech segment. For each vowel, the mean over all of the vowel instances is trimmed, with 95% of the data kept, in order to limit the influence of potential outliers. Then, the convex hull (i.e., a polygon fit that encompasses all of the data points) is calculated in order to represent the vowel space area. The convex hull is selected to represent the vowel space area in this work because it effectively captures the maximum area that the points in the vowel space span. Figure 9 depicts the vowel space in the three conditions for speakers A1 and A2 as defined by the largest-area polygon fit (convex hull) for the 4 tense and 6 lax vowels (95% trimmed means).

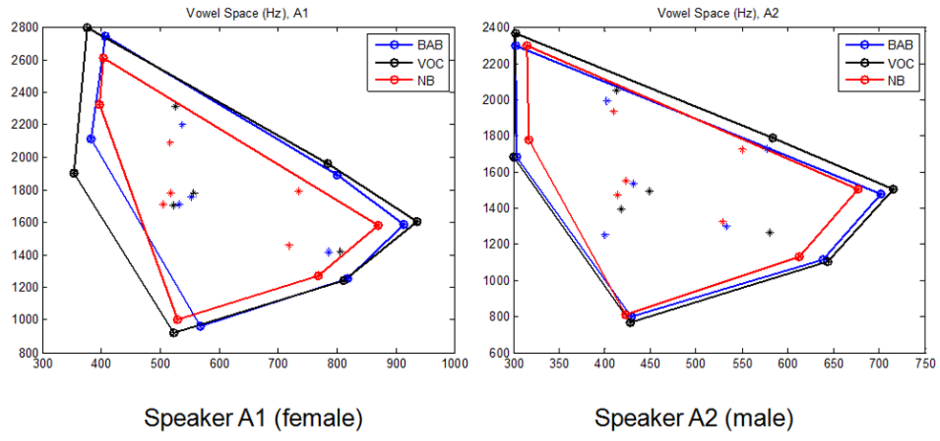


Figure 9: *Vowel space in the three conditions for speakers A1 and A2 as defined by the largest-area polygon fit (convex hull) for the 4 tense and 6 lax vowels (95% trimmed means).*

A per-vowel analysis was run on the measures using a mixed-model ANOVA, with vowel as a between-subjects factor, and condition (NB, BAB, VOC) as a within-subjects factor. The analysis showed a significant condition effect on all three vowels /i/, /ɒ/ and /ɔ/ for Speaker A1 ($p = 0.0398$) but no effect for Speaker B. So, for Speaker A, vowel space expands as follows: $NB < BAB < VOC$. Additionally, no significant interaction was found between vowel type and condition for either speaker.

4.6 Analysis of speech produced post communication breakdown

Sentences with more than two keywords incorrectly perceived were classified as having caused "communication breakdown". To find out whether there are significant differences between the pre- and post-breakdown sentences in terms of acoustic characteristics, for Speaker A1, acoustic analyses (i.e., sentence duration, LTAS at 1-3 kHz and 5-8 kHz, F0 median and range) were compared for breakdown sentences and post-breakdown sentences where all keywords were correct. In the BAB condition, there is a trend for longer sentence duration ($p = 0.163$) and significantly higher LTAS (5-8 kHz) post breakdown ($p < 0.05$). In VOC condition, effects were not significant but a trend for lower F0 median and higher F0 range post breakdown can be discerned. Although no correlation between sentence duration and communication effectiveness was found, a gradual increase in sentence duration was observed as time progressed in BAB condition for Speaker A1.

5 Examining prosodic differences between speech styles

The P8-Harvard corpus was also analyzed on time-domain, focusing on the number of pauses, mean word duration and the "rhythm" between the different speech styles.

5.1 Number and duration of pauses

In order to detect the number of pauses in the sentences of the whole P8-Harvard corpus, an algorithm was implemented to detect parts of speech signal with no proper speech content(NS, Not-Speech) such as pause between words, or even closure within stop consonants, etc. The silence detector relies on a *low-loudness detection* function based on the Perceptual Speech Quality measure. First the total loudness of the speech signal is computed by PSQ (ITU Standard REC-BS.1387-1-2001) and then the normalized loudness is computed dividing by the maximum loudness of the signal. A frame of the signal is considered NS if its normalized loudness is less than 15%. After cross-validation using a subset of files with manually-detected pauses (50 files from the P8-Harvard corpus) and it was found consistent. According to the linguistic context where the low-loudness part was located, the function could address the following type of Not-Speech:

- S: part of signal with loudness above threshold
- NS: generic low-loudness part of signal
- NS_{sil}: low-loudness part of signal at the beginning-end of the sentence
- NS_{sc}: low-loudness part of signal, which is part of a stop consonant inside a word
- NS_{iw}: low-loudness part of signal between two separate words (Inter-Word pause)
- $NS_{iwsc} : NS_{iw}$ in which the second word begins with stop consonant and therefore it is not possible to say if it is a pause or the closure of the consonant.

Applying the automatic detector to the P8-Harvard database, it was possible to compare the number of Not-Speech in different conditions. Table 2 contains the number of Not-Speech for each category, speaker and condition and Figure 10 has the average number of the total number of inter-word pauses per each utterance.

	A1			A2		
Type of NS	NB	BAB	VOC	NB	BAB	VOC
sc	437	492	*529*	433	470	*520*
iw	32	65	*155*	17	24	*75*
iwsc	176	209	*220*	130	162	*182*
sil	295	293	*298*	*277*	276	276
nc	1161	1386	*1615*	947	1059	*1247*

Table 2: *Number of instances of different type of NS*

The first results showed an increasing number of NS parts in the speech along with the difficulties in the communication. $\#NS_{VOC} > \#NS_{BAB} > \#NS_{NB}$ for both speakers, even though the male speaker tends to compensate less to the adverse conditions, as confirmed by other analysis.

A significant increase is worth to mention in the number of intra-words NS (NS_{iw}) between the VOC barrier and the other two conditions in both speakers as Figure 10 explicitly shows. This confirms that when the communication channel is really destructive and there is no direct feedback of it, the speaker focus the main part of his/her effort to greatly decrease the speaking rate.

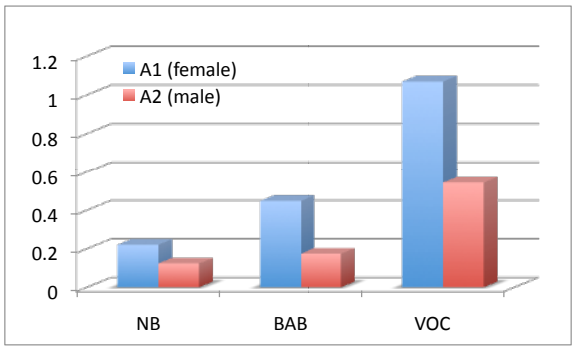


Figure 10: *Average number of inter-word pauses (NS_{iw}) for each utterance in different conditions.*

Further insight can be gained by looking at the durations of the different silence categories: Fig. 11 shows that, apart from leading/trailing silences (NS_{sil}), all types of silences undergo durational increase from NB to BAB to VOC, particularly the interword pauses (NS_{iw}). In contrast, speech part durations remain stable, highlighting a possible speaker strategy of reducing speech rate by detaching the words.

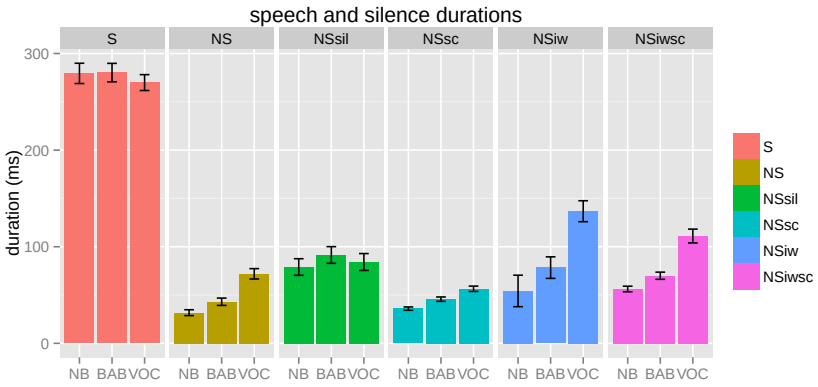


Figure 11: *Mean speech and silence durations for speakers A1 and A2 across NB, BAB and VOC. Errorbars are 95% confidence interval.*

5.2 Mean Word Duration analysis

The Mean Word Duration (MWD) for each type of condition was measured accurately using the silent detector, since the inter-word durations within utterances could be identified and subtracted to the word durations.

The results plotted in Figure 12 and Figure 13 display the change of duration in the VOC and BAB condition with respect to the No-Barrier condition during the experiments sessions. First observation is that all speakers elongate their speech production, especially in the worst condition (VOC barrier). This evolves along the sessions. However, this is not consistent between the two speakers for the BAB condition. Speaker A2 maintains mean word duration and mean content word duration stable. Speaker A2 was found to be less effective in the compensation, he slightly elongated the speech ($\sim 20\%$), only in the VOC barrier case but he didn't adjusted his speech any further. This lack of efficiency was confirmed by the amount of the errors the listener made which were much more compared to the errors he made during the session of speaker A1.

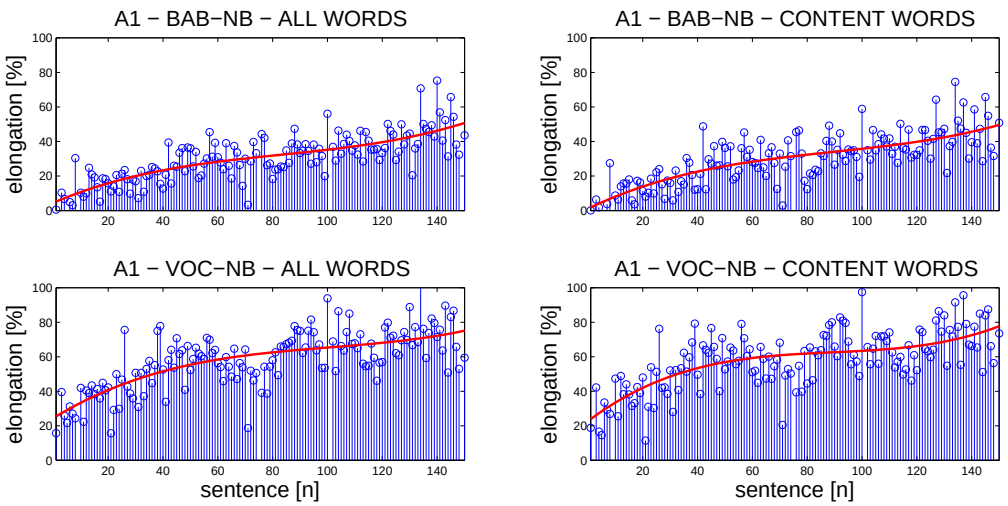


Figure 12: *Elongation strategy of female speaker A1 in the experiment sessions. On the left-hand side the mean word duration related to all words is shown, whereas on the right-hand side there is the mean content-word duration only.*

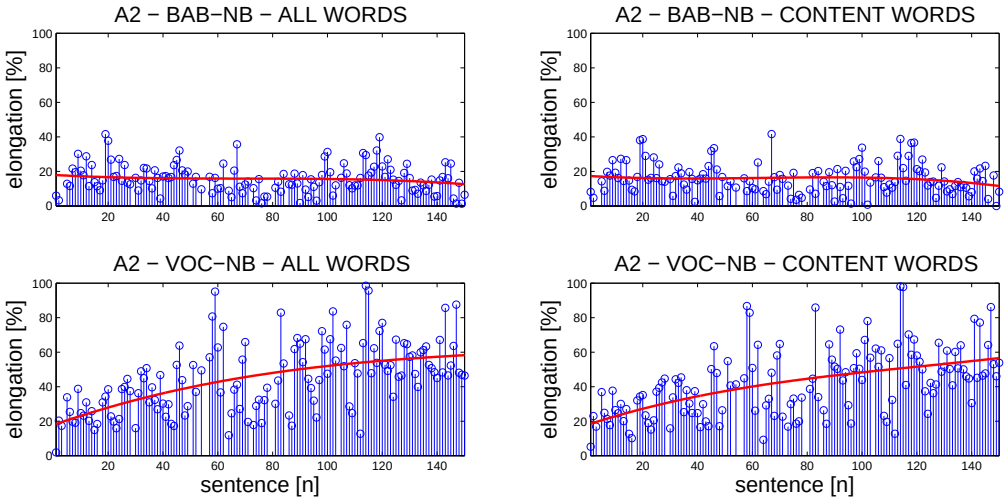


Figure 13: *Elongation strategy of male speaker A2 in the experiment sessions. On the left-hand side the mean word duration related to all words is shown, whereas on the right-hand side there is the mean content-word duration only.*

In Figure 12 and Figure 13 the red line is a 3rd-order polynomial curve that fits the data. Based on the shape of the line, three different stages emerge in all sessions, particularly for speaker A1 and the most stressful condition, i.e. the VOC barrier. At the beginning, the speaker starts with almost the same mean word duration as the NB condition, but as soon as he/she received intelligibility feedbacks from the listener, he/she increased the effort (i.e word duration) and hence a steep slope is seen at the beginning of the session. In the central part, the curve is flatter and the hypothesis is made that the current elongation is effective for the condition and the listener and no further adaptation is needed. In the final part, speaker A1 increased mean word duration further and it is hypothesised that she was trying to compensate the listener's tiredness, whereas, in the same conditions, speaker A2 seemed to cease making the effort to elongate, maybe due to a lack of motivation towards the end of the session.

Some correlations were investigated between the increase/decrease of mean word duration in a utterance and the number of listener's errors, but no clear relationship was found yet due to difficulties in comparing the two completely different data domains.

5.2.1 Rhythmogram analysis

The rhythmic patterns which differentiate barrier and no barrier speech was also investigated. For this task the rhythmogram [4] was employed. The Rhythmogram is a hierarchical representation of speech rhythm, from which one can extract the locations of relative prominences in the speech signal. This is achieved in a first step by computing auditory-based energy envelope with different time windows, and, by linking the peaks at different scales in a subsequent stage, enabling the identification of global (e.g., sentence-level) prominences (see Fig. 14). The detected prominences might undergo different

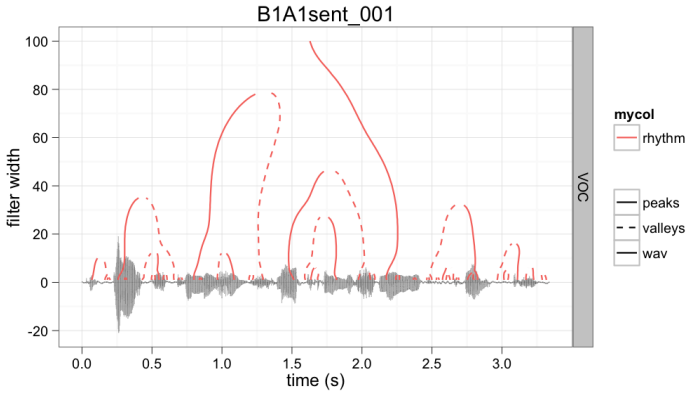


Figure 14: *Rhythmogram analysis for the sentence* "The birch canoe slid on the smooth planks". *Plain line stems identify relative prominences, dashed stems relative silences. Prominences and silences strength is determined by their highest value on the y-axis, and their location in the speech signal by their minimum value (smallest filter width).*

modifications by talkers in the reduction processes from clear to casual, and suggested a comparative analysis between clear and casual speech.

Using the manually annotated temporal mapping between clear and casual speech on a different Database (LUCID database), we assessed whether prominences and silences were treated differentially by talkers. Results on 69 pairs of matched casual/clear speech sentences showed that speech segments containing silences were significantly more compressed than prominences. ($p<.001$), Fig. 15. This shows that the nonlinearities observed in the temporal reduction from clear to casual can be explained by the rhythmic properties of speech: whereas silences appear to be suppressed from clear to casual, prominences tend to be preserved.

Given this result, prominence and silence locations were further characterized in terms of what sound class they fell in the P8-Harvard database comprising the NB and the two clear speech eliciting communicative barrier conditions BAB and VOC. The results presented in Fig. 16 show that most of the detected prominences fell into sonorant sound classes, i.e., vowels, nasals, and to a lesser extent, liquids. On the other hand, silences were found in stops, fricatives and annotated silences (occurring mainly in VOC condition, cf. Section 5.1). It should be noted here that the silences detected in the stop segments sound classes were mainly "low-level" silences in the rhythmogram hierarchy, and are not captured by

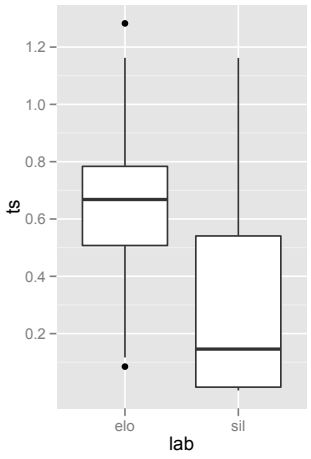


Figure 15: *Time scale factor of speech parts containing silences and prominences in clear speech.*

the global/salient silences detection. This analysis shows which segment would mainly benefit from an intrinsic time-scale modification based on the rhythmogram (cf. Section 6.2).

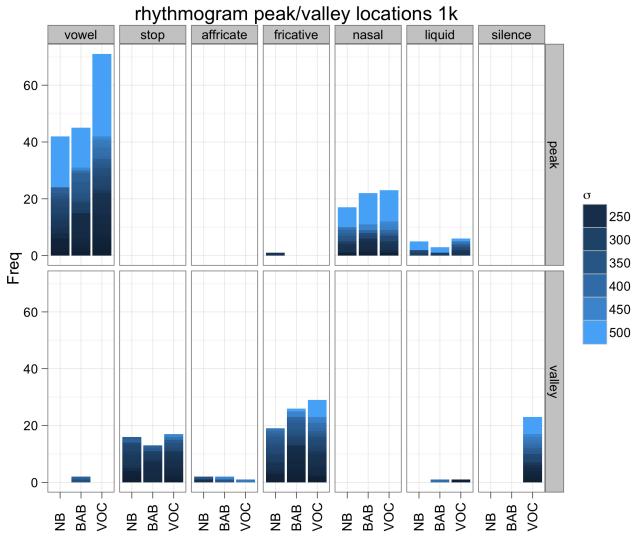


Figure 16: *Prominences and silences locations in sound classes in one non-barrier condition (NB) and two clear speech eliciting conditions (BAB and VOC). The filter width is controlled by σ: the higher the number, the more global the prominence / silence.*

These analyses provided a first pass characterization of the rhythmic properties of clear speech over casual or less clear speech styles. Future directions for assessing the specific places which differ between the speech styles could include acoustic analyses in the vicinity of detected prominences/silences, and global rhythmogram pattern analyses.

6 Proposed Time-scale modifications

Analysis of the P8-Harvard corpus showed a consistency of the speakers to elongate content words and add more pauses in the barrier cases. These adaptations result to a lower speaking rate of the speech signal. Therefore, two time-scaling modification techniques were developed in order to mimic the adaptations that speakers A1 and A2 make when they elicit clear speech in the barrier conditions. These time-scaling

techniques elongate the non-stationary parts of speech in order not to introduce artifacts to the speech signal and insert pauses to the signal. The first time-scaling technique is based on the Rhythmogram of speech and the second on the Perceptual Speech Quality Measure (ITU Standard REC-BS.1387-1-2001).

6.1 Perceptual Speech Quality Measure based Time-Scale Modifications

The Perceptual-Speech-Quality measure (PSQ) is used to elongate the stationary parts of casual speech and to define where to insert pauses to the signal. The Perceptual Speech Quality measure is based on the basic version of $ITUStandardREC - BS.1387 - 1 - 2001$, a method for objective measurements of perceived speech quality. It estimates features such as loudness and modulations in specific bands, in order to describe the input signal with perceptual attributes.

Two metrics of the PSQ model are used to detect the stationary parts of speech, where time-scaling can be applied: the perceived loudness of the signal in low bands and the loudness modulations in high bands. Analytically, PSQ estimates the perceived loudness on the low frequency bands (0-300Hz) of the signal, where unvoiced speech is less likely to be present. However, some voiced stop consonants, e.g. /d/, have high energy in low frequency bands. Time-scaling voiced stop consonants would cause distortion. Therefore, the loudness metric is not sufficient to decide which parts should be elongated. Then, another metric is introduced, namely the loudness modulations of high frequency bands (around 4000Hz). The loudness modulations in high frequency bands are strongly correlated with the non-stationarity of the signal and are able to detect voiced stop consonants. Therefore, the combination of the two metrics is proposed, called the Elongation Index (EI), defined as:

$$EI = \begin{cases} L-M, & L-M < threshold \\ -1, & L-M > threshold \end{cases} \quad (2)$$

where L is the average perceived loudness in low bands and M the loudness modulations in high frequency bands. EI is calculated for each frame of the signal. If EI exceeds a threshold then the frame is allowed to be elongated. The lower the threshold, the more likely is to capture non-stationary parts. EI does not depend on the energy of the signal and its threshold is defined between [1.3 - 1.4].

An example of how EI is calculated for a speech signal is shown on Figure 17. The speech signal depicted on Figure 17 corresponds to the phrase "made a sign." The loudness in low frequency bands, as calculated by PSQ, is depicted with a green curve whereas the modulations in high frequency bands are in red. Voiced phonemes like /ey/ and /e/ have high loudness on low bands and low modulations on high bands. In these cases, EI is above the threshold and these phonemes are allowed to be elongated (Figure 17b). Phoneme /d/, as a voiced consonant, has high loudness in low bands as well as high modulations in high bands, so it is not allowed to be elongated. For consonants like /s/ the loudness metric is lower than the modulation metric. Therefore, they will not be elongated either. Notice that the value of EI in Figure 17b is not important, rather, the sign of EI is taken into account.

Each frame that can be elongated is now the center of a window with duration 20msec. Then, for this frame, a time-scale factor of 120% is created. The time-scale factor for the total sentence duration consists of the time-scale factors only for the frames that will be elongated. The time scale factors for these voiced frames are given as input to WSOLA [5], which then time-scales the signal.

6.1.1 Pause Insertion

Pause insertion is also implemented using the PSQ model. The perceived loudness of the speech signal in the whole band is estimated. Then, loudness is normalized by the maximum loudness of the signal and on this loudness curve, the valleys that are 30% lower than the maximum loudness of the signal are detected. The valleys with very low values, less than 10% of the normalized loudness of the signal, can be considered silences. On the other hand, it is observed that the valleys that fall within the loudness interval (10%, 20%] usually are in the middle of word boundaries and are appropriate for inserting pauses without distorting the signal. The pauses that result from these valleys are called aggressive pauses to distinguish them from the pauses derived from the valleys with very low values of loudness (non-aggressive). The PSQ algorithm adds both non-aggressive and aggressive pauses to the signal. The reason for the distinction between aggressive and non-aggressive pauses is that the algorithm uses different techniques to do the insertion. First, the non-aggressive pauses are inserted on the signal by adding a constant pause of 90 ms duration. Then, in order to insert the non-aggressive pauses on the location where the signal has higher loudness, a pre-processing of the signal before and after the location of the valley must be made. The pre-processing involves a time-scaling of the signal around the location where the gap will be inserted, if this is allowed by the stationarity restriction. Then, after scaling, a

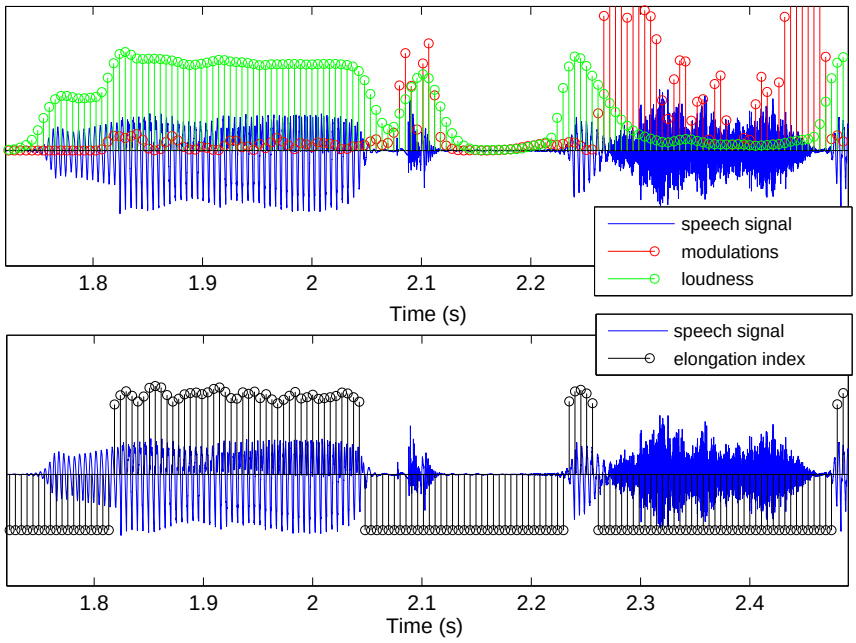


Figure 17: *Detection of non-stationary parts using PSQ model on the sentence "made a s(ign)" a) Loudness in low frequency bands and modulations in high frequency bands (top) b) Elongation index (bottom)*

hamming window is applied on the center of the valley so that the transition from speech to silence will be more smooth.

6.2 Rhythmogram-Inspired Time-Scaling and Pause Insertion

The speech rhythmogram has been proposed by Todd and Brown [6, 7] in order to model prosody perception. In order to generate the rhythmogram, the speech signal is first pre-processed to simulate the processing of the auditory periphery. In particular, the speech signal is first rectified and then raised to the one-third power. This processing approximates the loudness of the speech. Then, for the rhythmogram generation, multi-scale filtering is carried out by convolving the pre-processed speech with Gaussian windows of varying length in time. The peaks or prominences of the levels (corresponding to different Gaussian window lengths) are then linked in order to capture and visualize the overall rhythm of the speech. The following describes how this rhytmogram analysis of speech is used to inspire a time-scaling and pause insertion algorithm for speech.

Given the previously described observations on the differences between clear and casual speech, a PSQ-based algorithm for time-scaling and pause insertion was proposed. The rhythmogram provides a simple way to approximate the PSQ-based algorithm, in that it also elongates louder parts of speech while largely avoiding non-stationarities. Moreover, valleys in the rhythmogram level curves are used to detect where to insert pauses. Simplifying the processing by removing the need for calculation of the PSQ measure then frees up the rhythmogram-based approach (in terms of complexity) to provide additional pause enhancement using a WSOLA-based interpolation scheme. Explicitly, the rhthymogram-based time-scaling and pause insertion can be broken down and described in the following steps.

6.2.1 Pause Detection and Insertion

First, in order to approximate loudness, step 1 is to rectify the speech signal and raise it to the one-third power, mimicking processing in the auditory peripher. A "gross" Gaussian window (50msec length) is then convolved with the processed signal. The valleys in the resulting envelope then represent the longest pauses, or silences in the signal. This envelope is normalized so that its maximum value is one. The

location of the deepest valleys, defined as those more than 40% lower than the envelope maximum, are then used to indicate where zeros are inserted in the signal (see Figure 18). The length of the insertions are inversely proportional to the envelope valley depth, with the lowest valley being elongated the most (80msec).

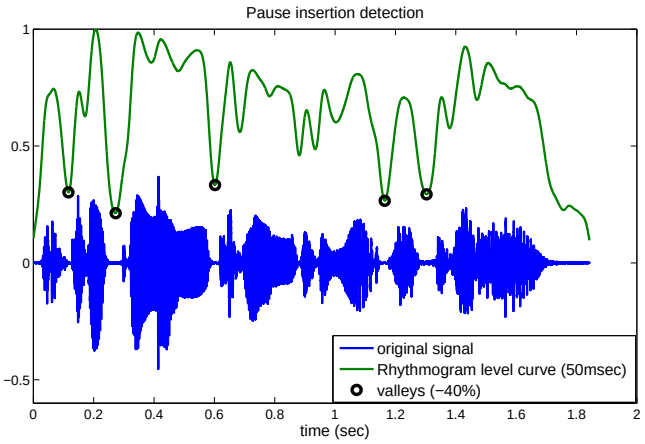


Figure 18: *Rhythmogram-based pause detection.*

6.2.2 Time-Scaling

A similar process to that used for the pause detection and insertion can also be used for time-scaling. In particular, the speech signal (with inserted pauses) is processed and the envelope (rhythmogram level curve) extracted in the same way. However, in this case, the time-scaling seeks to elongate prominences (peaks) and also silences (valleys) in the envelope. Like for the PSQ-based algorithm and uniform scaling described in this work, WSOLA is used for the time-scaling. Consequently, the normalized envelope from the rhytmogram level will determine the time-scaling factors that are input to WSOLA. In particular, the mean of the normalized envelope is first removed. The result is then rectified, so that the valleys become peaks. With this rectification, the parts of the envelope corresponding to transitions in speech (e.g., non-stationarities) lie near zero, as they have energy lower than the loudest parts of speech, yet higher than silences. The rectified envelope is then scaled by a maximum scaling factor, so that the time-scaling will not involve a factor larger than this amount. The scaling factors input to WSOLA are then one plus the scaled, rectified envelope. So, the non-stationary parts of speech will have a scaling factor near one and the rest of the speech will have a scaling factor above one, to elongate the signal, but below the specified maximum scaling factor (the maximum scaling currently limits the time-scaling factor to 2). Figure 19 shows an example of the input to WSOLA based on the rhythmogram-inspired time-scaling.

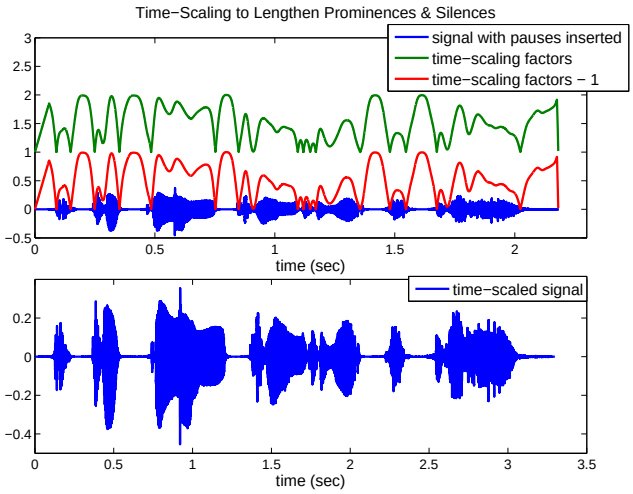


Figure 19: *Rhythmogram-based time-scaling.*

7 The GUI: XPlic8

As one of outputs of P8, XPlic8 is a MATLAB-based graphic tool for carrying out a series of analyses on single or batch of signals. It comprises of a set of functions for acoustic-phonetic measurement of speech, as typically used in speech science and phonetics research.

XPlic8 is able to perform seven acoustic-phonetic analyses and two visualization methods on sentence, word and phoneme levels. These are listed below:

- Analysis
  - Duration (s)
  - F0 median (Hz)
  - F0 range (Hz)
  - LTAS energy between a specified frequency range db SIL
  - Spectral tilt (dB/oct)
  - Vowel space (F1 (Hz), F2 (Hz))
  - Centre of gravity (Hz)
- Visualization
  - Source features analysis [8], [9]
    - LPC Spectrum
    - Harmonic-to-noise (HNR) ratio plot
    - Average glottal flow waveform
  - Vowel space plots
    - Plot of F1/F2 of tense vs. lax vowels
    - Plot of mean F1/F2 for all vowels
    - Plot of centre of gravity for /i/-/ɒ/-/ɔ/

Note that some analyses can only be performed on certain levels and also rely on the existence of corresponding annotation files for the signals. The detailed results from the analyses can be exported in plain text format that can be used as direct input for statistical applications such as SPSS or R for further analysis.

7.1 Analysis algorithms

The analysis algorithms developed during this project and incorporated in the GUI XPlic8 are described below.

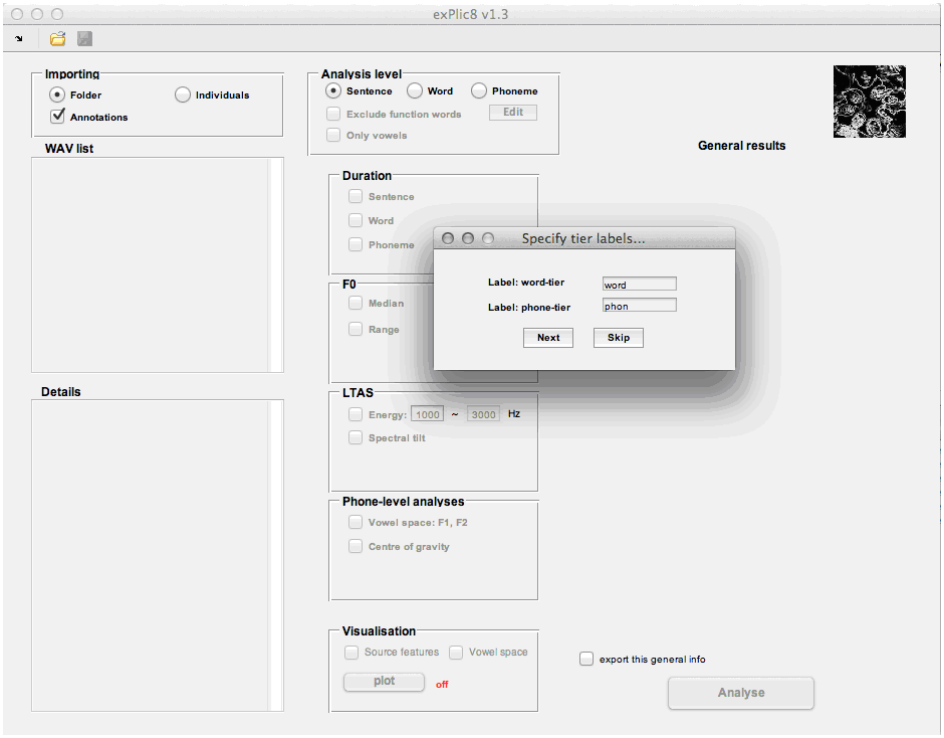


Figure 20: *The GUI XPlic8*

7.1.1 F0 estimation

A rough F0 trajectory prediction is performed prior to actual pitch detection. This is done in two stages: The first stage is to high-pass filter the speech signal in order to remove possible low frequency noise, followed by defining the rough F0 range. This is performed by using simple inverse filtering of the speech signal in order to remove most of the formants and then integrating the signal in order to get a signal close to glottal flow. This is done frame-wise with a 40-ms window. The rough fundamental period is estimated by evaluating the autocorrelation sequence of the signal and then finding the maximum peak that corresponds F0 between 50 and 500 Hz. Those frames with low energy or high zero-crossing rate (ZCR) are classified as unvoiced. F0 range is defined as:

$$F0_{min} = median(f_0)^{\frac{1.2}{5}} \tag{3}$$

$$F0_{max} = 2.2median(f_0) \tag{4}$$

The actual pitch detection takes place after the initial estimation of the F0 range. The analysis window size is adjusted to the estimated F0 range so that it is twice the lowest fundamental period ($2/F0_{min}$). The glottal inverse filtering method used in F0 estimation is iterative adaptive inverse filtering (IAIF) which estimates the glottal flow signal of the frame using linear prediction such that the fundamental period from the vibratory glottal flow waveform can be estimated. The fundamental period is estimated again finding the maximum peak of the autocorrelation sequence.

For post-processing, two highest peaks are saved: First, the post-processing involves forming a continuous trajectory from the two trajectories. This is based on the relative jump of the trajectories compared to a local F0 median. Second, 5-point median filtering is applied to smooth out outliers. Third, the unvoiced parts are set to zero based on the energy, ZCR, autocorrelation peak value, and gradient index. Fourth, the F0 trajectory is filtered with a 3-point medial filter. Finally, the median F0 is defined as the median of the non-zero values of the trajectory. The $F0_{min}$ and $F0_{max}$ are defined as the minimum and maximum non-zero F0 values of the trajectory.

7.1.2 LTAS energy in specified frequency ranges

The energy is computed as the intensity in SIL (sound intensity level) dB on the specified frequency range. The input sample is windowed with a 5-ms rectangular window without overlap and a 1024-length Fourier transform (using the FFT function) is computed for each frame. To obtain the normalized intensity for each frame, the energy in the specified frequency range is normalized by the length of the FFT, the length of the window (in samples) and the sampling frequency. Finally, the normalized intensities of all the frames are summed and the corresponding decibel value is computed by using the reference value $I_0 = 10e^{12}$.

7.1.3 Spectral tilt

The average spectral tilt is computed by fitting a regression line to 1/3-octave band energies of the LTAS (long-term average spectrum) in logarithmic scale. The LTAS is computed in 5-ms frames without overlap. For each frame, a 2048-length Fourier transform (with the fft function) is computed and the LTAS is obtained as the mean of the absolute values of the Fourier transforms over all frames. The average energy in the LTAS for each third-octave band is computed and normalized with the width of the band. These values are then transformed to logarithmic scale and a first-degree polynomial fit is estimated (using function polyfit). The average spectral tilt (in dB/octave) is three times the value of the first coefficient of the polynomial.

7.1.4 Vowel space (F1, F2)

The formant extraction tool returns the formant values in the middle point of the selected segment. It uses Praat [2] to extract the formant values for each consecutive frame in the selected speech segment and the cheapest paths through those values. Then, the values related to the centre of the time interval are chosen. This function returns formant info for every selected phone and this data is also used to plot the vowel space. Most of the analysis options are already optimised and cannot be changed: Time step = 0.01 s, Maximum formant number = 7, Number of paths to tracks = 5, Formant search range ceiling = 6500 Hz, Pre-emphasis filter lower limit = 50 Hz, Duration of the analysis window (0.025 s). For a detailed description of these parameters, please refer to the online Praat manual (Sound to Formant (Burg) and Formant Track)

7.1.4.1 Formant extraction The sound is re-sampled (Sound: Resample) to a frequency of twice the value of maximum formant and a pre-emphasis filter is also applied (Sound: Pre-emphasize (in-line)). For each analysis window, a Gaussian-like window is applied and the LPC coefficients are as per the algorithm by Burg, as (Childers, D.G., 1978) and (Press, W.H. et al., 1992). The number of "poles" in this algorithm is set as twice the maximum number of formants. The algorithm finds the best peaks in the selected range of frequency (between 0 Hz and the maximum formant value). Then, all formants below 50 Hz and above the ceiling minus 50 Hz are removed because very low frequency (near 0 Hz) and very high frequency (near the maximum) peaks cannot usually be associated with the vocal tract resonances and they are likely to be artifacts of the LPC algorithm.

7.1.4.2 Formant tracking After the formant candidate extraction, a tracking on these values is performed in order to rearrange the peaks to obtain the best formant tracks. This command uses a Viterbi algorithm with multiple planes and chooses the cheapest path through all the previously selected peaks (Formant Track). The cost function for one track (e.g. 2) with proposed values $F_{2,i}$ ($i = 1...N$, where N is the number of frames) is:

$$\begin{aligned} CostFunction = & \sum_{i=1}^{N} frequencyCost \frac{|F_{2,i} - referenceF_2|}{1000} \\ & + \sum_{i=1}^{N} bandWidthCost \frac{B_{2,i}}{F_{2,i}} + \\ & + \sum_{i=1}^{N-1} transitionCost |log_2 \frac{F_{2,i}}{F_{2,i+1}}| \end{aligned} \tag{5}$$

where frequencyCost, bandWidthCost, transitionCost, and referenceF2 values are fixed and all set to 1. Analogous formulas compute the cost of other tracks. The procedure will assign those candidates that minimize the sum of all-track costs.

7.1.5 Centre of gravity (CoG)

The Centre of Gravity is a measure of the spectrum energy distribution. The average spectrum on the speech segment is computed. It uses the Praat software [2]. Given the complex spectrum, S(f), f is the frequency, the CoG is computed by

$$\int_0^\infty f|S(f)|^p df \tag{6}$$

divided by the"energy"

$$\int_0^\infty |S(f)|^p df \tag{7}$$

The value of p is chosen to be 2. For further details please refer to the online Praat manual (Spectrum: Get the centre of gravity).

7.1.6 Source features

For details of F0 prediction refer to F0 estimation. The polarity is estimated by comparing the positive and negative energy of the glottal flow derivative signal. If the negative energy is greater, the speech signal most likely has positive polarity (and vice versa). After F0 and polarity detection, a suitable window size is selected for estimating the parameters ($3/F0_min$). Iterative adaptive inverse filtering (IAIF) is applied to the speech signal to separate the vocal tract transfer function and the voice source signal. Then, various parameters are extracted, such as:

- F0 and voiced/unvoiced decision [3]
- LPC and FFT spectra of voiced speech
- LPC and FFT spectra of unvoiced speech
- LPC and FFT spectra of vocal tract
- LPC and FFT spectral of voice source
- Speech energy
- Harmonic-to-noise ratio (HNR)
- H1-H2 value of the glottal flow signal
- Normalized amplitude quotient (NAQ)
- Individual glottal flow pulses and their average

The harmonic-to-noise ratio is evaluated by peak picking of the harmonics and then comparing the magnitude difference between the harmonics and the inter-harmonic valleys. These values are averaged to five equivalent rectangular bandwidth (ERB) bands. Normalized amplitude quotient is evaluated for each glottal flow pulse and thus averaged to one value for each frame. Finally, all the estimated unique glottal flow pulses are interpolated to constant length and averaged to estimate the average glottal flow waveform. Parameters are post-processed with median filtering. Statistics of the parameters are evaluated with 95% confidence intervals.

[3]Only available when single WAV file is selected and the analyses are performed on sentence level.

8 Summary and Conclusions

A new speech corpus, the P8-Harvard corpus, was linguistically and meta-linguistically annotated and acoustically analyzed. The aim was to identify which acoustic-phonetic characteristics differ between clear and casual speech then to modify casual speech to sound as intelligible as clear speech.

The P8-Harvard corpus contains, for each of two speakers, 150 sentences produced in a casual and two clear speaking styles. It is provided with word- and phoneme-level annotation, as well as pause annotations. Communication was harder in the communication barrier conditions, as shown by a decrease in keywords correctly transmitted, with the VOC condition being harder for both speakers. Acoustic-phonetic analysis revealed that sentence duration increased significantly in the barrier conditions, but that this was mainly due to an increase in pause duration, with a greater number of inter-word pauses seen in the more difficult (VOC) condition. Speakers also altered their F0 in the barrier conditions (higher F0 median in both conditions, broader F0 range in BAB condition only), and increased speech intensity (mid-frequency region), especially in the BAB condition. Speaker A1 hyperarticulated her vowels in the barrier conditions but no significant vowel space expansion was seen in male speaker A2. Evidence of communication-barrier specific strategies was seen. There was also evidence of differences in enhancement strategies across the two speakers for most dimensions.

Analysis of the P8-Harvard corpus showed a consistency of the speakers to elongate content words and add more pauses in the barrier cases. These adaptations result to a lower speaking rate of the speech signal. Therefore, two time-scaling modification techniques were developed in order to mimic the adaptations that speakers A1 and A2 make when they elicit clear speech in the barrier conditions. These time-scaling techniques elongate the non-stationary parts of speech in order not to introduce artifacts to the speech signal and insert pauses to the signal. The first time-scaling technique is based on the Rhythmogram of speech and the second on the Perceptual Speech Quality Measure (ITU Standard REC-BS.1387-1-2001). a set of evaluation experiments was prepared to evaluate the different modifications. The evaluation must be done by native listeners therefore no listening tests were conducted during the Enterface2012.

Finally, a significant outcome of P8 is XPlic8, a MATLAB-based graphic tool for carrying out a series of analyses on speech databases. It comprises of a set of functions for acoustic-phonetic measurement of speech, as typically used in speech science and phonetics research.

References

[1] V. Hazan and R. Baker. Does reading clearly produce the same acoustic-phonetic modifications as spontaneous speech in a clear speaking style? *DiSS-LPSS*, pages 7–10, 2010.

[2] Boersma P. and Weenink D. Praat: doing phonetics by computer [computer program]. 2012.

[3] IEEE. IEEE recommended practice for speech quality measurements. Technical Report No. 297, 1969.

[4] Neil P McAngus Todd and Guy J Brown. Visualization of Rhythm, Time and Metre. *Artificial Intelligence Review*, 10:253–273, 1996.

[5] M. Demol, K. Struyve, W. Verhelst, H. Paulussen, P. Desmet, and P. Verhoeve Author. Efficient non-uniform time-scaling of speech with WSOLA for call applications. *Proceedings of InSTIL ICALL2004 - NLP and Speech Technologies in Advanced Language Learning Systems*, 2004.

[6] N.P. Todd and G. Brown. A computational model of prosody perception. *ICLSP*, 10:127–130, 1994.

[7] N.P. Todd and G. Brown. Visualization of rhythm time and meter. *Artificial Intelligence Review*, 10:91–113, 1996.

[8] T. Raitio, A. Suni, J. Yamagishi, H. Pulakka, J. Nurminen, M. Vainio, and P. Alku. HMM-based speech synthesis utilizing glottal inverse filtering. In *IEEE Transactions on Audio, Speech, and Language Processing*, volume 19, pages 153–165, 2011.

[9] T. Raitio, A. Suni, H. Pulakka, M. Vainio, and P. Alku. Utilizing glottal source pulse library for generating improved excitation signal for HMM - based speech synthesis. *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4564–4567, 2011.

Inverse reinforcement learning to control a robotic arm using a Brain-Computer Interface

Laurent Bougrain[1,2], Matthieu Duvinage[3], *Research Fellow, FNRS,* Edouard Klein[1,4]

***Abstract*—The goal of this project is to use inverse reinforcement learning to better control a JACO robotic arm developed by Kinova in a Brain-Computer Interface (BCI). A self-paced BCI such as a motor imagery based-BCI allows the subject to give orders at any time to freely control a device. But using this paradigm, even after a long training, the accuracy of the classifier used to recognize the order is not 100%. While a lot of studies try to improve the accuracy using a preprocessing stage that improves the feature extraction, we work on a post-processing solution. The classifier used to recognize the mental commands will provide as outputs a value for each command such as the posterior probability. But the executed action will not only depend on this information. A decision process will also take into account the position of the robotic arm and previous trajectories. More precisely, the decision process will be obtained applying an inverse reinforcement learning (IRL) on a subset of trajectories specified by an expert. At the end of the workshop, the convergence of the inverse reinforcement algorithm has not been achieved. Nevertheless, we developed a whole processing chain based on OpenViBE for controlling 2D-movements and we present how to deal with this high dimensional time series problem with a lot of noise which is unusual for the IRL community.**

***Index Terms*—Inverse reinforcement learning, Brain-Computer Interfaces, Motor imaginery, Robotic arm**

I. Introduction

Brain-Computer interfaces (BCI) [1] interpret brain activity to produce commands on a computer or other devices like a robotic arm (see figure 1). A BCI therefore allows its user, and especially a person with high mobility impairment, to interact with its environment only using its brain activity.

A major difficulty to properly interpret the mental command lies in the fact that brain activity is very variable even if a particular task is reproduced identically. Beyond the noise acquired by the recording system, background brain activity, concentration, fatigue or medication of the subject are the source of this variability. This variability makes it difficult for the classifier to recognize the different mental commands. Specific preprocessings such as common spatial pattern filter [2] are useful to help distinguish the mental command. However, this effort is not always sufficient. It therefore becomes necessary to explore new solutions to address this variability.

[1]Université de Lorraine, LORIA, UMR 7503, Vandoeuvre-lès-Nancy, F-54506, France
[2]Inria, Villers-lès-Nancy, F-54600, France
[3]TCTS Lab, University of Mons, Belgium
[4] Supelec, Metz, France

Thus, it is now necessary to make decision systems able to deal with this variability. This is why some projects introduce a reinforcement learning in their BCI system such modifying the classifier [3]. We propose to use reinforcement learning in a broader context.

In this project we studied how a reinforcement learning can improve the control of a robotic arm. More precisely, the decision process will take into account a subset of trajectories specified by an expert and the position of the robotic arm in addition to the usual outputs of the mental commands classifier.

II. Methods

The goal of this study is to present the possible improvement on command recognition obtained by a post-processing performed by an inverse reinforcement learning algorithm. In this section, we first present the almost standard processing chain we used to obtain four different commands using motor imagery. Then we present how inverse reinforcement learning can help to better identify the mental order provided by the user.

A. A BCI system based on motor imagery

For controlling a neuroprosthesis of the upper limb several options are available nowadays. Firstly, the neural activity in the arm/hand area of the motor cortex can be directly recorded and decoded using invasive [4] or noninvasive electrodes ([5], [6]). But it is also possible using noninvasive electrodes to exploit various physiological phenomena such as sensorimotor rhythms, event-related desynchronization/event-related synchronization, event-related potential or steady-state visual evoked potentials. In particular, motor imaginary [7] can be used to control a 2D cursor ([8], [9]) or perform a 3D control [10]. They can even be combined in a hybrid BCI [11]. We selected motor imagery for several reasons: i) intending to produce a real movement is more natural for controlling a neuroprosthesis, ii) no additional device is needed to produce stimulations if used in a self-paced mode [12] iii) it has been already used successfully with healthy people [13] and patients [14] and iv) it can be used for rehabilitation [15]. Nevertheless, the number of commands is small (two or three usually); the information transfer rate is slow (1 action per 8s); and the accuracy is not very high (80 %).

We used motor imagery (MI) in a system-paced BCI. Having a self-paced BCI is not essential for this study and it is technically easy to switch from one mode to the other.

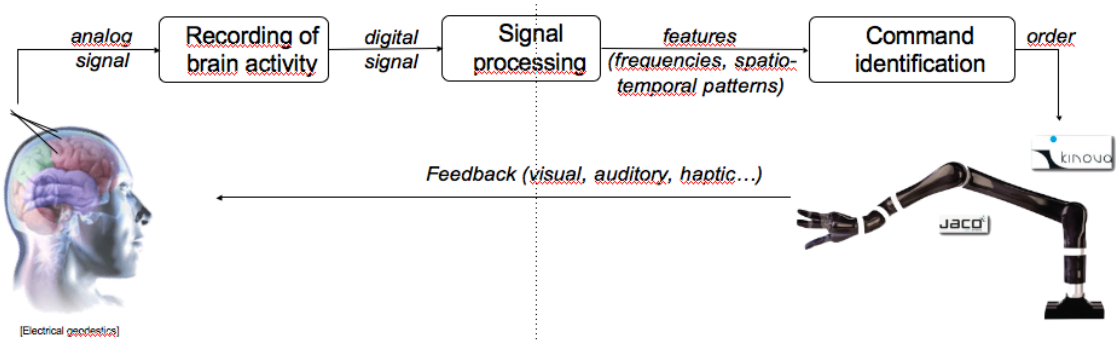


Fig. 1. The Brain-Computer Interface loop : from electroencephagraphic signals acquisition, feature extraction and classification to feedback. Our project will add a decision process based on an inverse reinforcement learning in the command identification module.

We defined a standard processing chain for motor imagery based on the parameters used for the Graz paradigm. We want to identify four commands corresponding to four motor imageries: left hand, right hand, both hands and feet. These four MI will allow us to control a robotic hand in a 2D horizontal space using respectively left, right, forward and backward commands [16].

We used a conventional montage for MI when applying a preprocessing based on common-spatial filters [17], [18], [2]. Then, among various possible classifiers to detect the MI [19], we selected linear discriminant analysis for its stability. More details are presented in the following sections.

1) Signal acquisition: We used a TMSi Refa amplifier with 32 EEG channels. We only selected 13 electrodes : Fz, FC5, FC1, FC2, FC6, C3, Cz, C4, CP5, CP1, CP2, CP6, Pz (see Fig. 2) located according to a layout 10/10 on a WaveGuard 32 channel sintered Ag/AgCl. This system use a AFz ground and a common average. We used a sampling rate of 512 Hz.

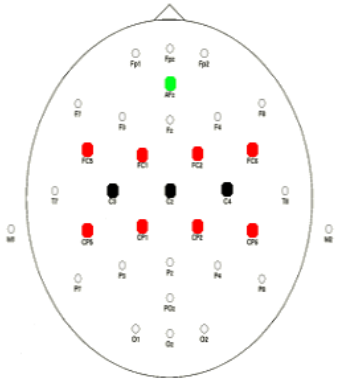


Fig. 2. Position of the selected electrodes for motor imagery of left hand, right hand, both hands and feet. The green electrode corresponds to the ground, the black ones are the main locations of our motor imageries and the red ones are useful for common spatial patterns.

2) Pre-processing: We used a 4th order Butterworth band-pass filter 8-30 Hz to only keep *mu* and *beta* bands.

Then we applied a Common Spatial Pattern (CSP). This filter takes into account the distribution of each class of a two-classes classification. The variance of the filtered signal is maximal for one class and minimal for the other class. Thus, we want to extremize using generalized eigen value decomposition:

$$J(w) = \frac{wX_1X_1^Tw^T}{wX_2X_2^Tw^T} = \frac{wC_1w^T}{wC_2w^T}$$

where X_i is the multichannel EEG signals from class i, C_i is the EEG spatial covariance matrix for class i and w is the spatial filter to optimize.

We obtained features $f = \log(wCw^T)$.

3) Motor imagery paradigm: Figure 3 presents our timing for motor imagery. Each session contains 20 trials per class. After the presentation of the cue, we analysis the signal for 3,5 seconds. The features are extracted for a 1-s period. We use a sliding window of 100 ms to repeat the analysis and confirm the decision of the classifier using a vote.

4) Classifier: For discriminating four motor imageries, we combined one-versus-all linear discriminant classifiers (one per class). In case of ambiguity, the longest distance to the separation plane shows the winner class.

5) Device: By default, the JACO arm can be controlled using a joystick. An API by Kinova is available to read sensors and send commands of movement for a specific direction and a specific duration. This API provides a virtual joystick. This mode of operation does not make it possible to specify the final position of the arm. Thus, interacting with the JACO arm via the API necessitates the definition of elementary movements (right, left, forward, backward, up.). The VRPN protocol already implemented in OpenViBE is a natural candidate to control the arm. Thus, we used a VRPN client/server using predefined action IDs which can be interpreted by the JACO arm as virtual joystick commands but sent through our application. The recording features also supports the recording of VRPN clients' commands.

B. Inverse Reinforcement Learning

Inverse Reinforcement Learning (IRL) is the problem of eliciting a succinct description of a task from demonstrations by an expert [20]. This succinct description of the task can then be used to train an agent in order to make it imitate the expert.

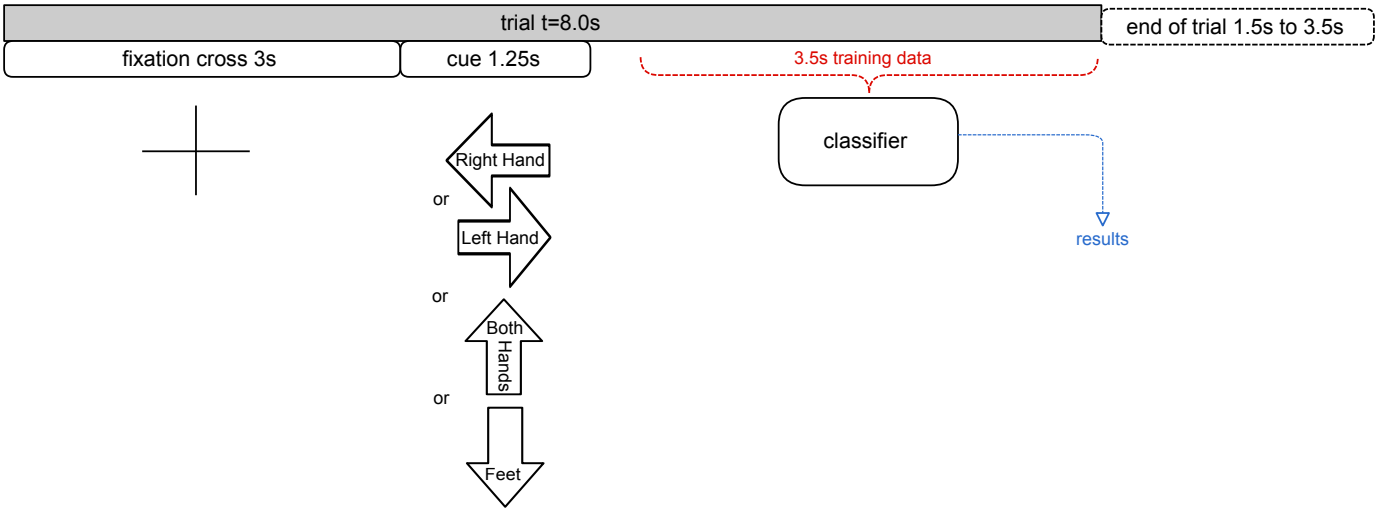


Fig. 3. Timing used for the motor imagery paradigm.

More formally, IRL assumes that an expert is acting optimally in an Markov Decision Process (MDP)[21] and seeks the reward function for which this expert is optimal. As noted in the existing literature, this is an ill-posed problem in the Hadamard sense. However, recent advances [22] in the domain may make solving the IRL problem on large or complex tasks feasible.

In our setting, we would like to use IRL to alleviate the problem of accuracy in order recognition from BCI signals. By using information about previously recognized commands and learning from human-labelled movement sequences, it should be feasible to gain a certain consistency in the overall arm movement. To put it in another way, after seeing a few examples of the arm moving in a direct manner from point A to point B, one is unlikely to admit a command that make the arm flail in seemingly random directions.

Using hand-labelled arm trajectories as expert demonstrations, we wish to extract a reward function that could be used to train an agent to recognize commands from BCI signals. The main challenges behind this task are the difficulty of finding a suitable MDP setting for casting the problem, the high dimensionality of BCI signals, the sparsity of data for both reward function inference and its optimization by an agent once the expert's actions have been analyzed by the IRL algorithm.

One of the main assumption of IRL is that the expert is acting optimally in an MDP with respect to an unknown reward function. Our goal when choosing a MDP setting for our experiment is to try to make that assumption hold. In previous test for the algorithm we used, the expert was explicitly created from a reward function and an MDP. Although the reward function was unknown to the IRL algorithm, it existed. Sharing the same MDP as the expert is one of the basic assumption made by the analysis of our algorithm. In this setting, however, the so-called expert is an omniscient agent as the path the arm followed was fixed in advance and the operator only had to follow it. There may or may not exist a MDP describing the process. We tried more than one characterization of the problem, discovering various flaws, and understanding better and better the subtleties of the exercise as we went on. This is described in the next section.

BCI signals typically are high dimensional time series with a lot of noise. From a signal processing perspective they are quite a challenge. This is very unusual for the IRL community who is more used to toy problems (although impressive applications have been published [23]) where the dimension is low and the observation perfect. IRL can be applied to partially observable environments, although it is not the direction we wish to take here as it has its own set of challenges, mainly related to computation cost explosion. The high dimensionality problem has been circumvented by the use of SCIRL, a new IRL algorithm that among other advantages is quite fast to run. The low signal to noise ratio, however, is at the heart of our problem and the very reason for the existence of this project. It raised its lot of problems when trying to come up with a reasonable MDP setting.

The model for our system being unknown yet (although modeling the brain have been promised over and over by sci-fi authors, it is not yet within the reach of a one-month project) we had to rely on sampling to make things work. This means that we had to rely on expert demonstration only to retrieve a reward function and to optimize it. Reward inference from expert data only is one of the marketed features of SCIRL. Having access to samples drawn by a random policy is one of the many ways to run a Reinforcement Learning (RL) algorithm, and the most accessible to us. The high practical cost of generating samples with a BCI prevented us from getting even that in the allocated timeframe.

To wrap up, although IRL may alleviate the accuracy problem in BCI driven settings, the many challenges this approach implies are far outside the comfort zone of the community.

III. Results

We installed OpenViBE, a user-friendly open-source tool for BCIs, on a windows XP system. This system supports both a

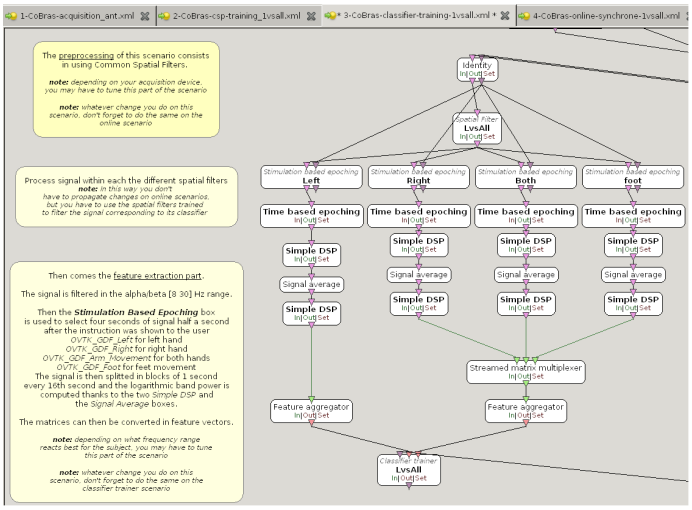


Fig. 4. OpenViBE scenario designed for training the one-versus-all classifiers.

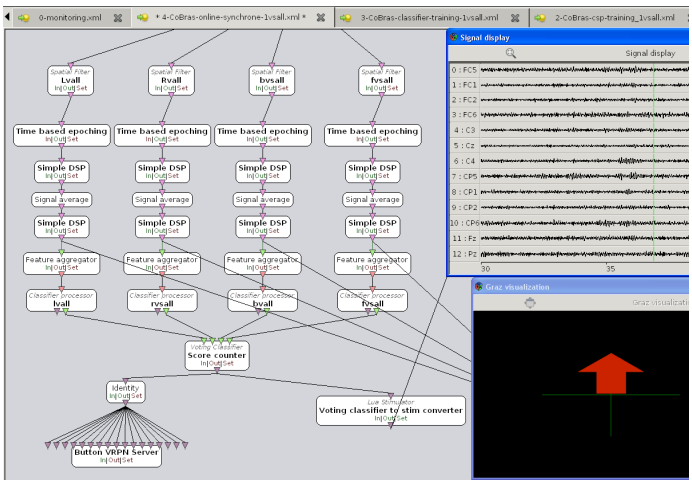


Fig. 5. OpenViBE scenario designed for on-line use. EEG signals are recorded, filtered, classified and one movement is sent to the robotic arm via the API.

JACO robotic arm driver and a Refa32 amplifier driver.

We built OpenViBE scenarii for i) signal acquisition ii) common spatial pattern filter training iii) classifier training and iv) offline use (see Fig. 4 and Fig. 5).

A. Standard motor imagery recognition

State-of-the-art similar results were obtained with imaginary and actual movements. The best combination strategy was the one-vs-all combined with a voting classifier. There were much less confusion and thus, better overall performance. It often happened that some classifier outputs had a very high confident level while the correct class was not represented. Confusion matrices were similar in both conditions (see Table I).

B. IRL

Let us disclose the end story immediately: not all challenges exposed earlier were overcomed.

TABLE I
CONFUSION MATRICE OBTAINED ON A TESTING SESSION.

Correct classes	*Predicted classes*			
	left hand	right hand t	both hands	feet
left hand	0.9	0	0.05	0.05
right hand	0.1	0.8	0.1	0
both hands	0.1	0.05	0.85	0.05
feet	0.1	0.1	0	0.8

The most hacky topic in the whole ordeal clearly was the composition of the state and action space of the MDP. Encouraging results were obtained on a simulation built to validate an initial approach. Sadly, this failed to generalize to the real thing as the real noise was much higher than modelized. A second, more sound approach was built, in which the state space directly consists in the output of the spatial filters and the last decision taken by the agent. This parametrization did not show any deep flaw and would be our goto parametrization if we are given the opportunity to work on this problem again.

The high dimensionality of the MDP was not a problem for our IRL algorithm, which was indeed able to infer a reward only from a few expert demonstration (corresponding to less than an hour of work for the operator).

Sadly, and this is the blocking point of the experiment so far, we were not able to train an agent on this reward. We need more data, specifically data sampled with a policy different from the expert's, in order to use the basic RL algorithm we tried to use [24]. We were thus not able to assess the quality of the found reward, although the fast convergence of SCIRL let us hope that it was good. We hope to solve this problem by either using less data greedy algorithm [25] or brutally generating more data (cumbersome for the operator). Another solution would be to use spatial filters able to deal with a displacement of the BCI, in order to allow the use of data from different sessions.

IV. CONCLUSION

We developed a whole processing chain using OpenViBE for controlling a robotic arm. According to the literature, we designed OpenViBE's scenarii (acquisition, filtering, classification and on-line use) based on a classic motor imagery paradigm. We selected four motor imageries (left hand, right hand, both hands and feet). They are respectively associates with 2D-movement (left move, right move, forward, backward). We used common spatial filters and one-versus-therest (linear discriminant analysis) classifiers. Our goal was not to improve the paradigm parameters but study how inverse reinforcement learning can help to select the right movement according to the classifier outputs and stored trajectories. Our classifier accuracy corresponds to the state-of-the-art. Thus, it is possible to control the Jaco to press a button. But, up to now, the IRL algorithm is not converging so cannot help to perform the right movement. Nevertheless, a significant analysis of the difficulties to apply IRL on high dimensional and noisy problem and ways to overcome them has been done.

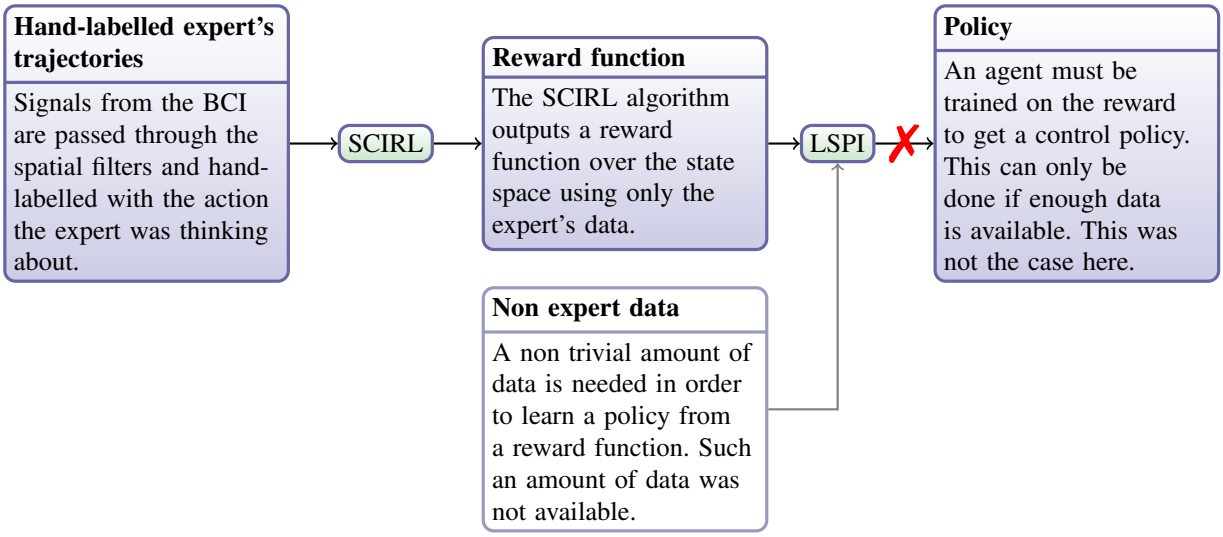


Fig. 6. Visual explanation of the IRL pipeline.

V. Perspectives

A deeper study is necessary for understanding the non-convergence of the IRL algorithm. If the IRL algorithm is robust enough, we will modify the processing chain to have a self-paced BCI. In the future, we also would like to use multiclass classifiers to avoid ambiguities due to the one-versus-therest approach. We would like to explore the tongue motor imagery to replace the both hands one. This choice avoids overlapped locations with the other motor imageries. We need to assess performance in offline and online conditions with a large population.

References

[1] J. R. Wolpaw, *et al.*, "Brain-computer interfaces for communication and control," *Clinical Neurophysiology*, vol. 113, no. 6, pp. 767–791, 2002. [Online]. Available: http://www.sciencedirect.com/science/article/B6VNP-45HFKTC-2/2/070472147433d00168e8d54909b982d2

[2] F. Lotte and C. Guan, "Spatially regularized common spatial patterns for EEG classification," in *International Conference on Pattern Recognition (ICPR)*, 2010.

[3] J. Fruitet, *et al.*, "Automatic motor task selection via a bandit algorithm for a brain-controlled button," INRIA, Research Report 7721, 2011. [Online]. Available: http://hal.inria.fr/inria-00624686/fr/

[4] L. R. Hochberg, *et al.*, "Reach and grasp by people with tetraplegia using a neurally controlled robotic arm," *Nature*, vol. 485, no. 7398, pp. 372–375, 05 2012. [Online]. Available: http://dx.doi.org/10.1038/nature11076

[5] T. J. Bradberry, R. J. Gentili, and J. L. Contreras-Vidal, "Reconstructing three-dimensional hand movements from noninvasive electroencephalographic signals." *The Journal of neuroscience : the official journal of the Society for Neuroscience*, vol. 30, no. 9, pp. 3432–3437, Mar. 2010. [Online]. Available: http://dx.doi.org/10.1523/JNEUROSCI.6107-09.2010

[6] G. M.-P. P. Ofner, "Decoding of hand movement velocities in three dimensions from the eeg during continuous movement of the arm," TOBI Workshop III, 2012. [Online]. Available: http://www.tobi-project.org/sites/default/files/public/Publications/TOBI-247.pdf

[7] G. Pfurtscheller, *et al.*, "Current trends in graz brain-computer interface (bci) research," *Rehabilitation Engineering, IEEE Transactions on*, vol. 8, no. 2, pp. 216 –219, June 2000.

[8] J. R. Wolpaw and D. J. McFarland, "Control of a two-dimensional movement signal by a noninvasive brain-computer interface in humans," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 101, no. 51, pp. 17 849–17 854, Dec. 2004. [Online]. Available: http://dx.doi.org/10.1073/pnas.0403504101

[9] H. Yuan, C. Perdoni, and B. He, "Relationship between speed and eeg activity during imagined and executed hand movements," *Journal of Neural Engineering*, vol. 7, no. 2, p. 026001, 2010. [Online]. Available: http://stacks.iop.org/1741-2552/7/i=2/a=026001

[10] A. S. Royer, *et al.*, "EEG control of a virtual helicopter in 3-dimensional space using intelligent control strategies." *IEEE Trans Neural Syst Rehabil Eng*, vol. 18, no. 6, pp. 581–9, 2010.

[11] P. Horki, *et al.*, "Combined motor imagery and ssvep based bci control of a 2 dof artificial upper limb." *Med. Biol. Engineering and Computing*, vol. 49, no. 5, pp. 567–577, 2011.

[12] G. Townsend, B. Graimann, and G. Pfurtscheller, "Continuous eeg classification during motor imagery-simulation of an asynchronous bci," *Neural Systems and Rehabilitation Engineering, IEEE Transactions on*, vol. 12, no. 2, pp. 258–265, June 2004. [Online]. Available: http://dx.doi.org/10.1109/TNSRE.2004.827220

[13] C. Guger, *et al.*, "How many people are able to operate an eeg-based brain-computer interface (bci)?" *IEEE Trans Neural Syst Rehabil Eng*, vol. 11, no. 2, pp. 145–7, 2003. [Online]. Available: http://www.biomedsearch.com/nih/How-many-people-are-able/12899258.html

[14] K. K. Ang, *et al.*, "Clinical study of neurorehabilitation in stroke using eeg-based motor imagery brain-computer interface with robotic feedback." *Conf Proc IEEE Eng Med Biol Soc*, vol. 1, pp. 5549–52, 2010. [Online]. Available: http://www.biomedsearch.com/nih/Clinical-study-neurorehabilitation-in-stroke/21096475.html

[15] V. Kaiser, *et al.*, "First steps towards a motor-imagery based stroke bci: New strategy to set up a classifier," *Frontiers in Neuroprosthetics*, vol. Special Topic "Future invasive and non-invasive Brain-Machine-Interfaces(BMI) for acute and chronic stroke", 2011. [Online]. Available: http://www.tobi-project.org/sites/default/files/public/Publications/TOBI-136.pdf

[16] M. Naeem, *et al.*, "Seperability of four-class motor imagery data using independent components analysis," *Journal of Neural Engineering*, vol. 3, no. 3, p. 208, 2006. [Online]. Available: http://stacks.iop.org/1741-2552/3/i=3/a=003

[17] T. Wang, J. Deng, and B. He, "Classifying eeg-based motor imagery tasks by means of time?frequency synthesized spatial patterns," *Clinical Neurophysiology*, vol. 115, no. 12, pp. 2744–2753, 2004. [Online]. Available: http://dx.doi.org/10.1016/j.clinph.2004.06.022

[18] A. Bashashati, *et al.*, "A survey of signal processing algorithms in bci based on electrical brain signal," *J Neural Eng.*, vol. 4, no. 2, pp. 32–57, 2007.

[19] F. Lotte, *et al.*, "A review of classification algorithms for eeg-based brain–computer interfaces," *Journal of Neural Engineering*, vol. 4, no. 2, 2007. [Online]. Available: http://stacks.iop.org/1741-2552/4/i=2/a=R01

[20] A. Ng and S. Russell, "Algorithms for inverse reinforcement learning," in *Proc. ICML*, 2000, pp. 663–670.

[21] M. Puterman, *Markov decision processes: Discrete stochastic dynamic programming*. John Wiley & Sons, Inc. New York, NY, USA, 1994.

[22] E. Klein, *et al.*, "Structured Classification for Inverse Reinforcement

Learning," in *European Workshop on Reinforcement Learning (EWRL 2012)*, Edinburgh (UK), 2012.
[23] P. Abbeel, A. Coates, and A. Ng, "Autonomous helicopter aerobatics through apprenticeship learning," *International Journal of Robotics Research*, vol. 29, no. 13, pp. 1608–1639, 2010.
[24] M. Lagoudakis and R. Parr, "Least-squares policy iteration," *The Journal of Machine Learning Research*, vol. 4, pp. 1107–1149, 2003.
[25] M. Geist and O. Pietquin, "Kalman Temporal Differences," *Journal of Artificial Intelligence Research (JAIR)*, vol. 39, pp. 483–532, October 2010. [Online]. Available: http://www.metz.supelec.fr/metz/personnel/geist_mat/pdfs/Supelec632.pdf

Laurent Bougrain is an associate professor at the university of Lorraine (France). He is a member of the Inria team NeuroSys dedicated to computational neuroscience at LORIA/Inria Nancy grand Est. He has been working for more than a decade on time series analysis with a focus on experimental data obtained during neuroscientific experiments. In recent years, he has dedicated his research to Brain-Computer Interfaces (BCI). He is working on template-based classifier for single trial detection using multichannel denoising techniques. He is the winner of the international BCI competition IV of the challenge about predicting the finger flexion from ECoG in 2008. He is currently working on a project on reinforcement learning to control a robotic arm and a wheelchair from EEG. He also collaborates to the worldwide BCI-software OpenVibe (http://openvibe.inria.fr). E-mail: bougrain@loria.fr, http://www.loria.fr/~bougrain.

Matthieu Divinage As a TIME student, Matthieu Duvinage holds an Electrical Engineering degree from the Facult Polytechnique of Mons (UMons, Belgium, 2009) and one degree from SUPELEC (France, 2009). He also holds a degree of fundamental and applied physics from Paris Sud XI Orsay (France, 2009) and a degree of management science from the School of Management at the University of Louvain (UCLouvain, 2011). His master thesis was performed at the Multitel research center (Mons, Belgium) and dealt with robust low complexity speech recognition using frame dropping based on voicing information and clustering techniques. He obtained an F.R.S-FNRS grant for pursuing a PhD thesis about the development of a lower limb prosthesis driven by a neural command in close partnership with the Universit Libre de Bruxelles (ULB).

Edouard Klein is a PhD student co-supervised by Yann Guermeur (ABC team, CNRS), Matthieu Geist (IMS research group, Suplec) and Olivier Pietquin (IMS research Group, Suplec and UMI 2958 (GeorgiaTech - CNRS)). The topic of his PhD is automatic feature selection in inverse reinforcement learning. He received the Electrical Engineering degree of PHELMA (Grenoble, France) in 2010. Since that, he has worked on reinforcement learning and learning from demonstration in the IMS research group of Suplec.