

© Presses universitaires de Louvain, 2005

Registration of copyright: D/2005/9964/22

ISBN : 2-87463-003-9

Cover :

Printed in Belgium

All rights reserved. No part of this publication may be reproduced, adapted or translated, in any form or by any means, in any country, without the prior permission of Presses universitaires de Louvain.

Distribution : www.i6doc.com, on-line university publishers

Available on order from bookshops or at

CIACO University Distributors

Grand-Place, 7

1348 Louvain-la-Neuve, Belgium

Tel. 32 10 47 33 78

Fax 32 10 45 73 50

duc@ciaco.com

Participant list eNTERFACE'05



PARTICIPANT LIST eNTERFACE'05
Summer Workshop on Multimodal Interfaces

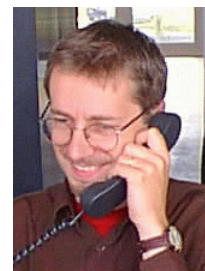
By the Similar Network of Excellence Consortium

Edited by Thierry Dutoit



eINTERFACE Workshops

Motivations - Reflexions



Prof. T. Dutoit

Initiator of the eINTERFACE concept and organizer of eINTERFACE'05

I take this opportunity to mention how the idea of eINTERFACE came to life in mid 2003, when the core committee of SIMILAR (the European Network of Excellence on Multimodal Interfaces), was busy establishing its list of workpackages. It is basically a three acts piece.

Act 1. I have been a researcher for 18 years now. It is therefore becoming hard to navigate in the "conferences" directory of my laptop. Small to big, short to long, close to far away, I think I have tried them all. One of them, however, will last forever in my memory as the most productive meeting I have ever attended. It was a summer school on Prosody, in July 1993, organized by the ELSNET (already a scientific network). I spent two weeks there, at UCL London, attending lectures and, more importantly, taking labs with my fellow PhD students from all over the world. I must say this is simply the place where I met most of my friends for life!

Act 2. In 1996, I had the opportunity to work for AT&T at Bell Labs for 1.5 years, in the TTS group. This was set about 2 years after I finished my PhD (i.e., 2 years after I had signed with Kluwer for writing the "3-months-of-work" book in TTS I took 3 years to complete; I finished it at AT&T...). It was clear to me that I was then about to meet the greatest gurus in speech processing (yet I had underestimated the number of famous people who were working in this lab), and that I would work with the best maintained software archive in the world (you snap your finger, and you get what you were looking for; this, I had overestimated...). I did meet all these people, and the atmosphere was such that meeting each other was really easy, but I also realized something I had never imagined: research in the US is a huge network thing. "Network" in terms of "you seldom work on your own on a problem", but also in terms of "Be with the network; the network will take care of you". In other words, research is very much advertised and supported by your employer, by all sorts of professional organizations, and even among the general public. Hence its dynamics.

Act 3. I was aware of the successful DARPA workshops on speech recognition organized yearly by Prof. Fred Jelinek at Johns Hopkins University. Funded by the Defence Agency (which implies a strong financial support), these workshops have progressively become a "must" for researchers in the field, who come from all around the world to participate. One of our researchers took part to it, and my ex-colleague Hervé Bourlard, now the Director of IDIAP in Switzerland, was an active member of it. I have always envied this event, and dreamt of finding money to organize something SIMILAR.

Thanks to EU financing, and with special care from SIMILAR, this dream has come true.

With its 55 researchers from 15 countries all around the world working together for four weeks on seven pre-selected projects, eINTERFACE'05 has been a total success.

Long life to eINTERFACE workshops!
See you next year in Dubrovnik for eINTERFACE'06!

T. Dutoit



eINTERFACE Workshops

Actively building the European Research Area

What are eINTERFACE workshops?

The eINTERFACE summer workshops (www.enterface.net), organized by the SIMILAR European Network of Excellence, are a new type of European workshops. They aim at establishing a tradition of collaborative, localized research and development work by gathering, in a single place, a group of senior project leaders, researchers, and (undergraduate) students, working together on a pre-specified list of challenges, for 4 weeks. Participants are organized in teams, attached to specific projects related to multimodal interfaces, working on free software.

eINTERFACE'05 was held at Faculté Polytechnique de Mons, Belgium, in July-August 2005 (see next article). The eINTERFACE'06 workshop will be organized in Dubrovnik, Croatia, in July-August 2006.

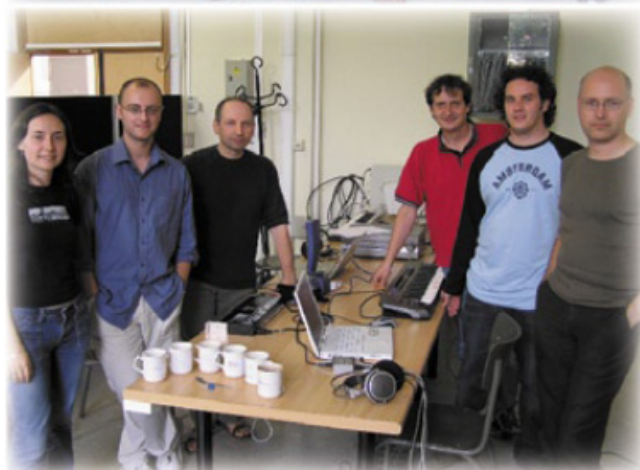
What do eINTERFACE workshops produce as results?

At the end of the workshop, a public presentation day is organized, in which the team leaders explain and demonstrate the results of their project. A press conference is also organized, to maximally publicize the event.

All results, codes and data, are then made publicly available with an MIT-like open source license.

Last but not least, the workshop proceedings are produced 6 weeks after the end of the workshop, in which each team contributes a 15pp. paper on the project they had to study, the related state-of-the-art-of-the art, the problems encountered, and the solution(s) proposed and implemented.

But still more importantly, eINTERFACE workshops create a real transfer of know-how among participants, who continue to work together after the workshop has closed. They actively contribute to building the European Research Area, by establishing a tradition of



Three of the seven eINTERFACE'05 teams

localized collaborative research.

The eINTERFACE funding model

No funding is provided by the organizers for researchers, but no registration fees are asked for either. Participants therefore have to pay for their travel, lodging, and catering expenses, using their SIMILAR finances or other EU, national, or regional funding. Catering and lodging is available from the University organizing the workshop, at minimal student rates. Some grants are available from scientific societies.

A limited number of undergraduate students (typ. 10) are also selected (based on their CV and recommendations from professors), whose travel and accomodation expenses are paid by the organizers.



The Summer Workshop on Multimodal Interfaces July 18th - August 12th, Faculté Polytechnique de Mons, Belgium

The eINTERFACE'05 workshop was the first of a series of eINTERFACE workshops, held at Faculté Polytechnique de Mons, Belgium, from July 18th to August 12th, 2005. Following the general organization process of eINTERFACE workshops, seven projects were first selected on the basis of an international call for projects. From this list of projects, a call for participation was then launched internationally, and participants were selected, on the basis of their CV and potential input in the projects. This call resulted in the selection of 55 researchers from 15 countries all around the world (but mostly Europe), organized in 7 teams.



Each team work for a complete month on one of the following 7 challenges:

1. **Combined Gesture-Speech Analysis and Synthesis** (Coordinators : Profs. Murat Tekalp, Engin Erzin, Yucel Yemez, Mehmet Emre Sargin, Koc University Multimedia, Vision and Graphics Lab, Istanbul).
This project included the preparation of a database, the study of correlations between speech features and gesture units, the modeling of gesture units, and the adaptation of units for specific speakers.
2. **Multimodal Caricatural Mirror** (Coordinator : Prof. B. Macq, UCL Louvain La Neuve).
This project aimed at creating a caricatural mirror where people could see their own emotions amplified (image+speech) by an avatar, on a wide screen facing them. It includes

Realization of 1st prototype

Dialog controller

Fusion redundancy separation

Network messages (text)

Image feedback

Text feedback

Voice feedback

High frequency?

Both eyes closed?

Significant rotation?

Voice?

ECG analysis

GSR analysis

Eye lines detector

Head rotation estimation

Mouth opening detector

ECG

GSR

Video

Audio

FACE 2005

The 5th Annual International Conference on Face and Gesture Recognition

- The goal was to build a virtual musical instrument driven by EMGs, EEGs, Heart beats, video, etc.

- (Coordinators: Profs. Laurent Bonnaud & Alice Caplier, INPG_LIS, Grenoble; Prof. B. Macq, TELE Lab, UCL Louvain La Neuve).

5. **Multilingual Multimodal Biometric Identification/Verification** (Coordinator : Prof. Yannis Stylianou, University of Crete)

6. **Speech Conductor** (Coordinator : Prof. C. D'Alessandro, LIMSI-CNRS, Paris)

7. **A Multimodal (Gesture+Speech) Interface for 3D Model Search and Retrieval Integrated in a Virtual Assembly Application** (Coordinator : Prof. Dimitrios Tzovaras ITI-CERTH, Thessaloniki).

All projects were presented to the press on August 12th; this press conference led to multiple appearances in the national TV and radio news, as well as to several articles in national newspapers.

vi

The eINTERFACE'05 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*
Ferran Marques, *Univertat Politècnica de Catalunya PC, Spain*
Michael Schnaider, *Zentrum für Graphische Datenverarbeitung e.V, Germany*
Michael Strintzis, *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 eINTERFACE'05 Organizing Committee

Coordination: Prof. T. Dutoit
Catering and lodging: D. Wynsberghe - S. Devuyst
Web management: C. Ris - N. D'Alessandro
Finances: D. Wynsberghe - S. Devuyst
Registration management: F. Séverin
Publications : Dr. B. Arslan - R. Sebbe
Publicity: S. Fereira - Y. Hiernaux
Tutorials: B. Bozkurt
General intendance: V. Gaudissart - C. Thillou
Social activities : L. Couvreur

The eINTERFACE'05 Sponsors

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



eNTERFACE 2005

PROJECT REPORTS

Project 1: Combined Gesture-Speech Analysis and Synthesis	1
Project 2: Multimodal Caricatural Mirror	13
Project 3: Biologically-driven musical instrument	21
Project 4: Multimodal Focus Attention Detection in an Augmented Driver Simulator	34
Project 5: Multilingual Multimodal Biometric Identification/Verification	44
Project 6: Speech Conductor	52
Project 7: A Multimodal (Gesture+Speech) Interface for 3D Model Search and Retrieval Integrated in a Virtual Assembly Application	62

Combined Gesture-Speech Analysis and Synthesis

Mehmet Emre Sargin, Ferda Ofli, Yelena Yasinnik, Oya Aran,
Alexey Karpov, Stephen Wilson, Yucel Yemez, Engin Erzin and A. Murat Tekalp

Abstract—Multi-modal speech and speaker modelling and recognition are widely accepted as vital aspects of state of the art human-machine interaction systems. While correlations between speech and lip motion as well as speech and facial expressions are widely studied, relatively little work has been done to investigate the correlations between speech and gesture.

Detection and modelling of head, hand and arm gestures of a speaker have been studied extensively in [3]-[6] and these gestures were shown to carry linguistic information [7],[8]. A typical example is the head gesture while saying "yes". In this project, correlation between gestures and speech is investigated. Speech features are selected as Mel Frequency Cepstrum Coefficients (MFCC). Gesture features are composed of positions of hand, elbow and global motion parameters calculated across the head region. In this sense, prior to the detection of gestures, discrete symbol sets for gesture is determined manually and for each symbol, based on the calculated features, model is generated. Using these models for symbol sets, sequence of gesture features is clustered and probable gestures is detected. The correlation between gestures and speech is modelled by examining the co-occurring speech and gesture patterns. This correlation is used to fuse gesture and speech modalities for edutainment applications (i.e. video games, 3-D animations) where natural gestures of talking avatars is animated from speech.

Index Terms—Gesture Recognition, Keyword Spotting, Audio-Visual Correlation Analysis, Prosody Analysis, Gesture Synthesis.

I. INTRODUCTION

The role of vision in human speech perception and processing is multi-faceted. The complementary nature of the information provided by the combinations of visual speech gestures used in phoneme production (such as lip and tongue movements) has been well researched and shown to be instinctively combined by listeners with acoustic and phonological information to correctly identify what is being said. Visual information can also provide listeners with certain aspects of paralinguistic knowledge about a speaker, helping them to be located in space, as well as supplying information regarding their age, gender and emotional intent.

The project detailed in this report, seeks to perform a preliminary exploration of potential correlations between non-facial gestures and speech, with the goal of providing natural gesture patterns for the task of artificial gesture synthesis. The project consists of a number of inter-connected modules, sketched below:

- Audio-Visual Analysis: A fine-grained examination of the audio-visual data is carried out, seeking to identify

salient gesture patterns and relevant speech phenomena for potential correlation;

- Preparation of Training Data: Careful selection and preparation of samples of gestures and keywords for training automatic speech and gesture detectors;
- Head and Hand Tracking;
- Keyword Spotting;
- Gesture Recognition;
- Gesture Synthesis and Animation;

II. MOTIVATIONS AND INITIAL OBSERVATIONS

A primary motivation of the work presented here was to identify natural classes of gestures that conveyed real linguistic meaning, that is, to identify gestures or groups of gestural patterns that could be clearly correlated with information conveyed in the speech signal. Once identified, these classes would be used to synthesize "natural" gesture patterns using an animated stick figure, given an input speech signal. The work detailed below is intended to be a preliminary investigation and so is restricted to analyzing gestures in a limited but gesture-rich task. An audio-visual database was prepared, comprising 25 minutes of video data. A single native speaker of Canadian English was recorded, providing directions to a number of known destinations in response to questions given off camera.

An initial informal analysis was carried out, in order to ascertain potential lexical candidates that had recurring patterns of significant gestures. This involved close viewing of the video data by two investigators with experience of gesture identification and speech annotation. Initial observation highlighted three candidates, "left", "right", and "straight", for further study. The three lexical items were chosen as they showed a high co-occurrence with periods of significant manual gestural activity. Furthermore, they had a high distribution throughout the database indicating a potentially rich source of data for analysis. It was informally noted that 28 instances of the candidate "left", appeared to be accompanied by some sort of gesture. Similarly 31 occurrences of "right" had accompanying gestures, while "straight" had associated gestures 32 times throughout the database. Other candidate words included "across", "no", and "down", but these were dismissed as having too few gesture-marked occurrences (8, 8, and 6 respectively).

III. DATABASE AND TOOLS

For the database we used a 25 minute video recording of the experiment in which a subject is asked to provide another person with directions from one location to another. The speaker is a native speaker of Canadian English and is familiar with all the locations asked.

This report, as well as the source code for the software developed during the project, is available online from the eNTERFACE'05 web site: www.enterface.net.

This research was partly funded by SIMILAR, the European Network of Excellence on Multimodal Interfaces, during the eNTERFACE05 Workshop in Mons, Belgium.



Fig. 1. Examples of "Straight" and "Left" Gestures



Fig. 2. Examples of "Nod" and "Tilt" Gestures

A. Gesture Analysis

The video was viewed in Virtual Dub 1.6.9 which allowed the normal speed playback accompanied by sound, as well as stepping frame by frame at a rate 25 fps. Based on initial observation of directional words and gestures that were salient in the video, the following hand gestures were manually labelled:

- right and left gestures - the right or left hand turns to make a 90° angle with the arm, pointing to the right for right gesture, or to the left for left gesture;
- straight - the subject starts with her hands in parallel, palms facing each other, fingers directed up, and moves the hands away from the body by extending her elbows. The finishing position is with hands parallel, palms facing each other, fingers pointing away from the subject's body.

For each gesture identified as a right, left, or straight gesture, we noted the frame numbers corresponding to the initial and the final hand positions of the gesture movement. If the subject's hands stayed in the final hand position for several frames without movement, only the first frame was noted and recorded as the end of the gesture.

Ten examples of each gesture were snipped from the video in order to serve as training data for the gesture recognizer. Six minutes of the video were labelled for the right, left, and straight gestures without sound and used for analysis of correlation with speech which will be explained later.

Head gestures were examined and seemed to correlate with prominences in speech. Since evidence for correlation between sharp head movements and prosodic events in speech has been presented in gesture literature previously [2], we have decided to narrow down our investigation of head gestures to nods and head tilts. These gestures were manually labelled without listening to the audio and following the criteria:

- nod - the head comes down with chin closer to the body and sharply comes back up;
- tilt - the head rotates right or left 45° from its natural vertical position.

Again, ten examples of each gesture were saved as clips and served as training data for the gesture recognizer. Two minutes of the video were labelled for the nods and tilts for analysis of correlation with speech prominences.

B. Speech Analysis

The speech was investigated using the 25 minute .wav file corresponding to the video. The phonetics annotation toolkit Praat 4.3.19 was used to view the waveform, pitch,

spectrogram, and intensity of the sound. All 25 minutes were manually transcribed for words "right," "left," and "straight" using spectrogram and waveform to identify precisely beginnings and ends of these words. In the 25 minutes the word "left" was said 28 times, the word "right" - 29 times, "straight" - 46, giving us a database of 103 keywords.

As we have previously mentioned, while a potential correlation between head gestures and certain lexical items may exist, (nods linked with words pertaining to agreement or assertion, tilts with words associated with hesitation), initial informal analysis, driven in part by previous research outlined in the literature [1], implied a strong correlation between head gestures and prosodic events known as pitch accents. The Tone and Break Indices (ToBI) prosody labelling convention was chosen to mark prominences in speech [19]. In order to establish an initial working hypothesis, an experienced ToBI labeller marked 2 minutes of speech for pitch accents and phrase boundaries. This two-minute section of the sound file corresponded to the video segment previously labelled for head gestures. Our labels deviated from the ToBI notational convention in that the pitch accents were marked as intervals spanning the whole accented syllable, rather than single point events. Within the 2 minute segment, there were 122 identified pitch accents.

IV. CORRELATION ANALYSIS

After some manual labels have been provided for speech and gesture events, several correlation analysis were conducted in order to provide justification for two hypotheses:

- directional hand gestures are closely correlated with the identified lexical candidate tokens, such as "left", "right" and "straight";
- sharp head movements, such as nods and tilts, are closely correlated with speech prominences marked as pitch accents.

In this section we will describe the correlation analysis procedure and results for both of these two hypotheses.

A. Directional Hand Gestures

Within the six minute video fragment labelled for directional hand gestures and speech keywords as described in the Database section, 23 gestures were manually identified and fell into categories summarized in numbers below as well as in the Figure 3 with percentages. Of the 23 gestures, 15 were direct matches with the candidate words "left", "right", and "straight", meaning that there was some degree of temporal

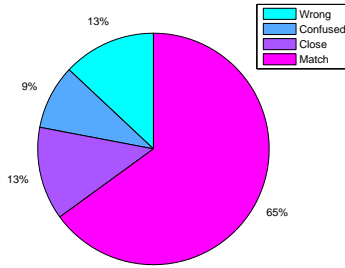


Fig. 3. Summary of Directional Word-Gesture Alignment

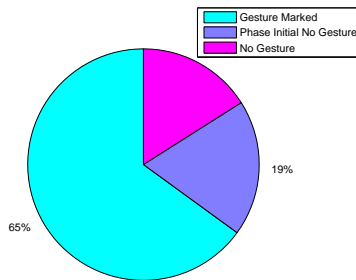


Fig. 4. Summary of Identified Pitch Accents

overlap between the gestures and corresponding keywords. A further 3 gestures were deemed to be "close", being in the judgement of the labeller associated with one of the candidates, but with the phase of the gesture narrowly falling outside of the duration of the word (2 were off by one frame, the third missed by 360 ms). Of the remaining 5 gestures, 3 were wrongly identified as being related and 2 were designated as "confused", meaning that speaker has correctly used the gesture to indicate going left, right or straight, but that the phase of the gesture overlaps with another candidate word, usually being used in a different context. For example, the phrase: "Take a left and go *straight* down that street" had two accompanying left hand gestures. The first overlapped with the keyword "left" and was deemed a match, the second with the keyword "straight" and was marked as "confused".

B. Head Gestures

The two-minute sample file labelled for prosody and sharp head movements was found to contain 122 pitch accents and 81 head gestures: 66 nods and 15 tilts. Of the 122 pitch accents, 79 or 64.75% overlapped with a head gesture, either a nod or a tilt. It is worth noting, that from the 43 pitch accents that did not overlap with a head gesture, 23 or 53.5% were phrase initial accents, which are known to be problematic in prosody labelling (see Figure 4). Often phrase initial stressed syllables are misidentified as pitch accents due to the fact that both pitch accents and phrase initial syllables are accompanied by tense voice quality [20].

If we disregard the 23 phrase initial syllables that were labelled as accents, only 20 of the 100 pitch accents identified

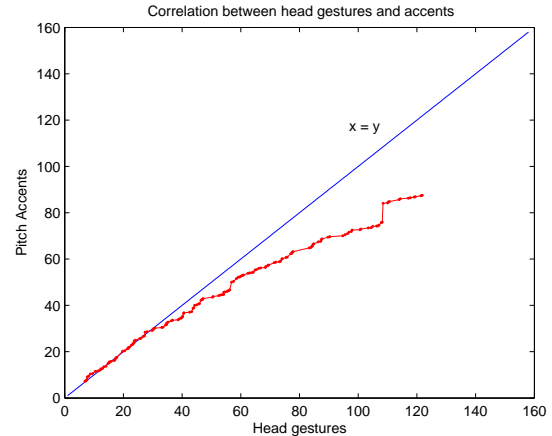


Fig. 5. Head Gesture and Accent Correlation Graph

did not overlap with a sharp head movement, that is 80% of remaining accents co-occurred with a head gesture.

The 79 accents that overlapped with a nod or a tilt were also examined for temporal correlation with the relevant head gesture. Time-stamp labels of the accented syllable were compared to the start and end time-stamps of the overlapping gesture using the statistical test of Pearson's correlation ran in Matlab. The correlation test produced Pearson's correlation coefficient $r=0.994$, which implies almost perfect correlation. The corresponding correlation plot can be seen in figure 5.

V. RECOGNITION OF AUDITORY EVENTS

The ultimate aim of the recognition process described in the following section is to automatically detect selected auditory events, namely the keywords, "left", "right", and "straight", as well as pitch accents. Detected instances of the chosen events will act as cues to animate the stick figure with correlated gestures.

A. Manual Labelling

In order to supply high-quality training data for the automatic keyword detector (Section V-B.1), the labels of the three keywords for a 20 minute portion of the sound file were used. Since a silence detector was also being trained, a number of examples of silence were also identified and labelled. All remaining non-keyword and non-silence portions of the 20 minute segment were presumed to be "garbage".

B. Automatic Labelling

Unlike in the keyword spotting procedure, the manual prosodic labels for the two-minute segment in the database were not intended to act as a training set for an automatic detector. Instead, they served as a "gold-standard" against which we could measure the effectiveness and accuracy of our automatic pitch accent detection strategies.

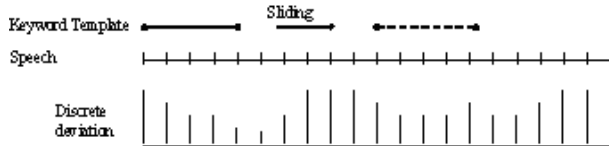


Fig. 6. Obtaining acoustic deviation function for a keyword

1) *Keyword Spotting*: The part of research during the project was connected with realization of special speech recognition system (keyword spotting system) for finding in unknown speech the words which describe the direction: left, right and straight. In our case it should be speaker-dependent because we have only one voice. Such system can be based on known techniques for speech recognition: dynamic time warping or statistical modelling.

a) *Dynamic time warping based keyword spotting system*: Dynamic time warping method is the approach, which allows finding an optimal match between two given sequences (e.g. time series). The dynamic programming algorithm is usually used for searching the optimal match. This method was firstly applied for automatic speech recognition in the 60's for isolated word recognition. Among the advantages of this method the following can be selected: easy realization, the stage of training of acoustical models is not required as well as any necessity to prepare the training speech data. During the eINTERFACE'05 workshop the original method for keyword spotting was realized using C++. This method uses analysis in sliding window for comparison of keyword template with fragment of speech with calculation of acoustic deviation function along the speech utterance for each keyword [10]. The keyword template is the most typical pronunciation of keyword by the speaker. Also several pronunciations of each keyword can be used for analysis of the speech. The input signal from wave file enters into the module of parametrical speech presentation. In this module the sequence of digital samples is divided into speech segments. And a vector of parameters is calculated for each such segment. For parametrical representation we used the Mel Frequency Cepstral Coefficients (MFCC). The calculation of speech parameters is fulfilled by the parametrization module HCopy included in Hidden Markov Toolkit (HTK). Then the each parameterized keyword template ("LEFT", "RIGHT", "STRAIGHT") is shifted (slide) along the speech signal with some step and keyword template is compared with fragment of speech of the same length by dynamic programming (DP) method with calculation of deviation estimation between template and speech fragment in sliding window. Figure 6 shows the process of sliding DP-analysis. The word template slides along input signal with slide step and the DP-deviation between the template and signal part is calculated for every step forming the discrete deviation function. The smaller DP-deviation between the keyword template and signal part the more probability of this keyword appearance. At that the sliding step from 1 segment of speech till several speech segments can be applied. To select hypothesis of keywords in three streams of deviation functions (Figure 7) we use the thresholds which depend on the length

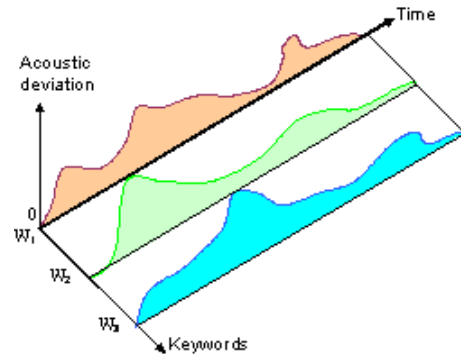


Fig. 7. Sliding analysis in the system

Recognized keywords	Missed words	False alarms
33	2	22

TABLE I

KEYWORD SPOTTING PERFORMANCE USING SLIDING ANALYSIS

of each template. Changing the thresholds we can manage the performance of the system and find some balance between word detection accuracy and number of false alarms made by the system. On the output of sliding analysis algorithm we combine the outputs of each stream and form the time-stamps for keywords in analyzable speech. For testing the system we used speech of one speaker with duration 330 seconds (5,5 minutes). In this speech there exist 35 keywords and this fragment contains about 600-700 continuously pronounced words. This fragment also was manually labelled, but it is required only for evaluation of the results of the system. The results of usage of this method are presented in Table I.

We used three criteria for evaluating the system: amount of properly recognized keywords in test speech, amount of missed keywords in test speech and amount of false alarms at analysis of speech. Thus the method showed 94,3%(33 of 35) in accuracy of keyword spotting and the same time gives about 3,6% (22 of 600) false alarms during analysis the speech. These results are not well enough but taking into account that the method does not require the construction and training of the models of words it can be used in some application areas for keyword spotting task.

b) *Hidden Markov Model based keyword spotting system*: At present the Hidden Markov Models (HMM) are most popular technology for statistical speech modelling and processing for diverse domains (not only for speech processing). It is difficult to realize effective system for statistical modelling using HMM is short time and therefore we used free available toolkit for training the HMMs. As the base technology for development of keyword spotter we have chosen the Hidden Markov Toolkit (HTK) developed in Cambridge University Engineering Department. HTK is free available toolkit which can be downloaded in Internet [11] and source code in C is available. Among the advantages of HTK the following should be noted:

- World recognized state-of-the-art speech recognition system

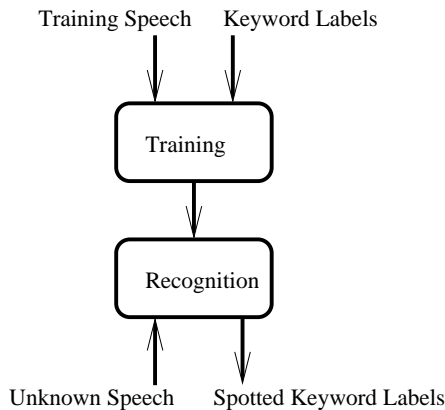


Fig. 8. General structure of system using HTK

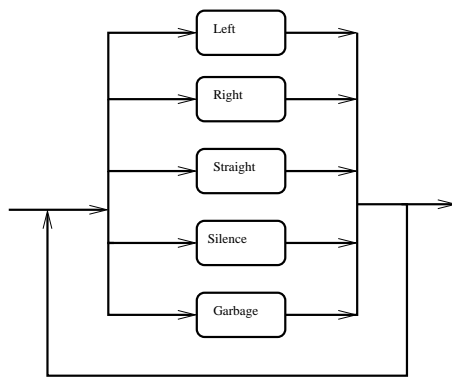


Fig. 9. Task Grammar

- Support a variety of different input audio file formats
- Support different feature sets
- Support almost all common speech recognition technologies.

The modelling of speech by HMM includes two stages (Figure 8):

- 1) Training the HMM using the database of training speech which was manually labelled
- 2) Testing this system by test speech.

The stage of models training includes the following steps:

- Definition of dictionary (lexicon of the task)
- Definition of task grammar - Preparing training speech data
- Coding the speech data (feature extraction)
- Definition of topology of HMMs (prototypes)
- Creating initial HMM models
- Re-estimation of HMMs parameters using speech data
- Mixture Splitting

At first the grammar for speech recognition should be defined. We used HMM for three keywords: "LEFT", "RIGHT" and "STRAIGHT" as well as defined the models for "SILENCE" (it is signal without any speech but with background noise only) and for "GARBAGE" (it is any other speech). This approach is similar to the approach described in [10]. The grammar for our task is shown on the Figure 9.

20 minutes of manually labelled speech was used for training the keyword spotting. Each keyword was pronounced

in training speech at least 30 times. The labels "left", "right", "straight" and "silence" with corresponding time of beginning and time of ending were set during manual analysis of training speech. The labelling was made using software Praat 4.3 but the output format of Praat with labels was not suitable directly for HTK processing because Praat uses the timestamps in seconds but HTK requires timestamps in 100 ns items. The special software was developed during the project for converting labelling data from the Praat format into HTK format. The feature extraction was performed using HTK configured to automatically convert input Wav files into vectors of parameters. As set of features we used Mel Frequency Cepstral Coefficients + delta coefficients + acceleration coefficients. Each feature vector includes 39 components: MFCC - 13, delta coefficients - 13, acceleration coefficients - 13. The next step in training stage was the definition of prototypes for HMMs. The parameters of prototype are not important, its purpose is to define the model topology. We used the left-right HMM with continuous observation densities in HMM. The number of states for each keyword depended on the number of real phonemes in each word. Thus for "LEFT" and "RIGHT" we used 14 states and for "STRAIGHT" - 20 states. This amount is calculated as number of phonemes in pronunciation of keyword multiplied by 3 and plus 2 states intended for concatenation of models. The prototypes for "SILENCE" and "GARBAGE" include 5 states each. Then the initial HMM models were created using the training speech and labels manually made in this speech. The re-estimation of parameters was performed using standard Baum-Welch algorithm for continuous density HMMs. After this step we obtained the trained HMMs for all words in our task. For recognition and testing the system we used 5,5 minutes of other speech of the same speaker. The recognition is performed by the Viterbi algorithm, the usage of N-best list on the output of the algorithm in this task is not possible. Thus we analyzed only an optimal hypothesis of speech recognition. According to the first experiments keyword spotter was able to find almost all keywords in test speech, but it gave many false alarms (above 30%). To decrease the number of false alarms we tried to apply the threshold for acoustical estimates but it did not give the acceptable results. The best results of system performance were obtained using mixture splitting and applying the multi Gaussian HMMs. In HTK the conversion from single Gaussian HMMs to multiple mixture component HMMs is usually one of the final steps in building a system. It allows creating more precise models for available training data. The number of mixture components in HMMs is repeatedly increased until achievement of the desired level of performance (keyword spotting accuracy and amount of false alarms).

The Figure 10 shows the dependence of keyword spotting accuracy and inverse rate of false alarms (100% - % of false alarms in speech). It can be seen that increasing the number of Mixtures of Gaussians we decrease the number of false alarms, because we tune our models better to the available training data and create more precise HMMs. But after some point the keyword spotting accuracy are decreased. It can be explained by amount of available pronunciations of keywords in training speech. Of course using other set of training speech

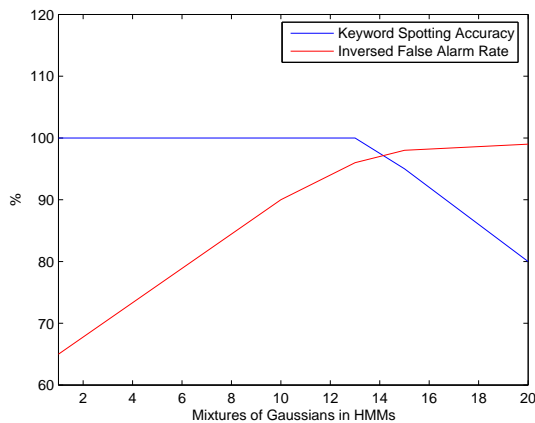


Fig. 10. Dependence of keyword spotting accuracy and inversed false alarm rate from the number of mixtures of Gaussian distribution in HMMs

Recognized keywords	Missed words	False alarms
33	2	10

TABLE II

KEYWORD SPOTTING PERFORMANCE USING HIDDEN MARKOV MODELS

these results will be changed.

Thus the best results on system performance were achieved by using 10-15 Gaussian mixtures in models. Table II shows the best results of keyword spotting by Hidden Markov Models based approach where the balance between keyword spotting accuracy and false alarm rate was found.

These results mean that we have 94,3% (33 of 35) in accuracy of keyword spotting and 1,6% (10 of 600) false alarms during analysis the test speech. These false alarms can be partly explained by specificity of pronunciation of the speaker. For instance sometimes in continuous real speech she pronounced the keyword "STRAIGHT" as "s t r i t" that is the same pronunciation as for the out-of-vocabulary word "STREET" or in her speech the keyword "RIGHT" was pronounced like "r e i". Thus after comparison of the Table I and Table II we have chosen the second version of keyword spotter based on Hidden Markov Models for the joint multimodal system in the project. The methods showed almost the same keyword spotting accuracy but the second approach was better in such criterion as false alarm rate.

2) *Prosodic Event Spotting*: Motivated from the correlation between accents and head movements, we propose an automatic methodology to extract accents from the speech signal. Proposed methodology uses pitch contour and intensity values as features. Pitch contour and intensity values have a frame rate of 100 samples per second. Pitch contour is extracted from the speech signal using autocorrelation method described in [9]. In order to extract intensity values of speech signal, the values in the sound are first squared, then convolved with a Kaiser-20 window with side-lobes below -190 dB. The effective duration of this analysis window is $3.2/(100Hz)$, where $100Hz$ is selected as minimum pitch frequency. The duration of analysis window should guarantee that a periodic

Recognized Accents	Missed Accents	False Alarms
68%	32%	25%

TABLE III

ACCENT DETECTION PERFORMANCE

signal is analyzed with a pitch-synchronous intensity ripple not greater than our 4-byte floating-point precision.

Algorithm first detects high intensity speech regions which are above the threshold $t_s = 48dB$. Median filter is used to smooth out small peaks from detected speech regions. Connected component analysis is applied to output of median filter in order to extract significantly long accent candidate regions. For each accent candidate regions, related pitch frequency sequence is investigated to eliminate non-accent regions. Number of peaks in the pitch contour for non-accent regions are usually few (0-2) or many (8-15). Therefore, the accent candidate regions that contain few or many peaks are eliminated and remaining regions are selected as accent regions.

The performance of proposed accent detector is determined using first 2 minutes of the database. Accent detector detected 85 accents out of 125 and the number of false alarms are 32. Performance rates of the accent detector can be seen in Table III.

VI. RECOGNITION OF GESTURAL EVENTS

In this section we present a framework for gestural event detection. Proposed framework can be divided into three tasks:

- 1) Manual Labelling of Gestures
- 2) Automatic Recognition of Head Gestures
- 3) Automatic Recognition of Hand Gestures

In the recognition phase, HMM based gesture recognizer is used and the HMM for each gestural event is trained using the manually labelled gestures.

A. Manual Labelling of Gestures

In order to train the automatic gesture detector and hand motion modeler, proper 10 examples for each gesture are labelled manually. Since all of the gestures are not well prepared gestures, elimination of non proper gestures is necessary.

B. Automatic Recognition of Head Gestures

In this section we present a methodology for head gesture recognition. Proposed methodology consists of three main tasks which are tracking of head region, extraction of head gesture features and recognition of head gestures based on Hidden Markov Models (HMM). Optical flow vectors calculated on head region are used to estimate new head position. New estimate of head region is corrected using skin color information. Head gesture features are extracted by fitting global head motion parameters to optical flow vectors. HMM is applied for recognition of gestures given the gesture features.



Fig. 11. Initial Head Region

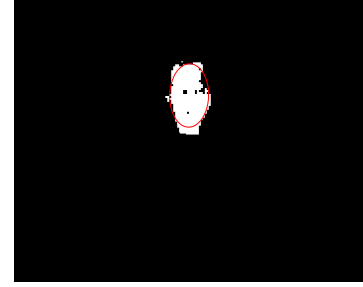


Fig. 12. Skin Region and Fitted Ellipsoid

1) *Initialization of Head Tracker:* Consider the first frame of the frame sequences. Since we do not have any prior knowledge about the initial position of head in the image, one should exhaustively search for face in the initial frame. For this purpose, boosted Haar based cascade classifier structure is used. Proposed object detector has been initially proposed by Viola [13] and improved by Lienhart [14]. The classifier is trained with positive and negative examples which are a few hundreds of sample views of face with size $M \times N$ and arbitrary images of the same size respectively.

Classifier consists of several simpler classifiers (stages) that are applied subsequently to a region of interest until at some stage the candidate is rejected or all the stages are passed. Classifiers at every stage of the cascade are complex themselves and they are built out of basic classifiers using one of four different boosting techniques which are Discrete Adaboost, Real Adaboost, Gentle Adaboost and Logitboost.

Trained classifier can be applied to a test image of the same size to determine whether applied image shows training object or not. One can find training objects of the same size on a whole image by running classifier on overlapping search windows across the image. In order to find same object with different sizes in other words to make classifier scale invariant, whole image can be scanned with different sized classifiers. Note that, once the classifier is trained using a specific sized object, the size of classifier is easily resized without training another classifier with different sized objects. Sample initial head position found by boosted Haar based cascade classifier can be seen in Figure 11.

2) *Extraction of Skin Blobs:* Skin blobs are extracted using color information. Here we assumed that distribution of Cr and Cb channel intensity values which belong to skin regions is normal. Thus the discrimination function can be defined as Mahalanobis distance from sample point \mathbf{x} to (μ_t, Σ_t) . Where \mathbf{x} is vector containing Cr and Cb channel intensity values $[x_{Cr}, x_{Cb}]^T$ and μ_t and Σ_t are the mean and variance of training skin regions respectively. Sample skin blob extracted can be seen as white area in Figure 12.

3) *Fitting Ellipsoid to Skin Blob Boundaries:* Suppose there are m points on the contour of a skin blob. Let $\mathbf{B} = [\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_m]$ where $\mathbf{x}_i = [x_{1,i}, x_{2,i}]^T$. Then center of the ellipsoid is given by $\mu_e = E\{B\}$. Principal axis and length of each principle axis will be the eigenvector and square root of eigenvalues of $\mathbf{B}\mathbf{B}^T$ respectively. Ellipsoid fitted to skin blob

can be seen in Figure 12.

4) *Optical Flow Observations:* Optical flow vectors will be calculated on the initial head region which was obtained in the previous step. Most trackable n points are selected on the initial head region by considering cornerness measure. Optical flow vectors are calculated on these n points which have the highest cornerness measure. Cornerness measure ρ can be defined as the minimum eigenvalue of covariance matrix of derivative image over the neighborhood S .

$$M = \begin{bmatrix} \sum_S \left(\frac{\partial I(x_1, x_2)}{\partial x_1} \right)^2 & \sum_S \frac{\partial I(x_1, x_1)}{\partial x_2} \frac{\partial I(x_1, x_2)}{\partial x_2} \\ \sum_S \frac{\partial I(x_1, x_1)}{\partial x_2} \frac{\partial I(x_1, x_2)}{\partial x_2} & \sum_S \left(\frac{\partial I(x_1, x_2)}{\partial x_2} \right)^2 \end{bmatrix} \quad (1)$$

$$\rho = \min(\text{eigval}(\text{cov}(M))) \quad (2)$$

Hierarchical Lukas-Kanade technique is applied to find motion vectors on n points. Implementation and technical details of algorithm can be found on [15].

5) *Estimation of Head Position:* Once the optical flow vectors are obtained on n points global motion parameters are fitted to n optical flow vectors calculated at n points. Estimation of global head motion parameters are given in Section VI-B.6. Center of search window is warped using global head motion parameters. Where warped search window will be the estimated position of head region in the next frame.

6) *Head Motion Feature Extraction:* Let optical flow vectors calculated at n points $[\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n]$ are represented with $\mathbf{d}_i = [d_{1,i}, d_{2,i}]^T$ where $\mathbf{x}_i = [x_{1,i}, x_{2,i}]^T$. Global motion parameters $[a_1, a_2, \dots, a_8]$ should satisfy the equation:

$$X = \begin{bmatrix} 1 & x_{11} & x_{21} & 0 & 0 & 0 & -x_{11}^2 & -x_{11}x_{21} \\ 0 & 0 & 0 & 1 & x_{11} & x_{21} & -x_{11}x_{21} & -x_{21}^2 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & x_{1n} & x_{2n} & 0 & 0 & 0 & -x_{1n}^2 & -x_{1n}x_{2n} \\ 0 & 0 & 0 & 1 & x_{1n} & x_{2n} & -x_{1n}x_{2n} & -x_{2n}^2 \end{bmatrix} \quad (3)$$

$$a = \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_7 \\ a_8 \end{bmatrix} \quad \text{and} \quad d = \begin{bmatrix} d_{1,1} \\ d_{2,1} \\ \vdots \\ d_{1,n} \\ d_{2,n} \end{bmatrix} \quad (4)$$

$$\mathbf{X}\mathbf{a} = \mathbf{d} \quad (5)$$

Since \mathbf{X} is tall system is overdetermined. Therefore one can find the solution using least squares and the least squares solution is given by:

$$\tilde{\mathbf{a}} = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{d}. \quad (6)$$

Note that, origin is selected as image center because the matrix $(\mathbf{X}^T \mathbf{X})$ is rank deficient if we select image origin as upper left corner. Note also that the more region of interest is far away from origin, the more the condition number is.

7) *Head Gesture Recognition and Gesture Spotting*: During the initial observation of the database, two head gestures (*down* and *tilt*) are identified as correlated with the pitch accents. The purpose of head gesture recognition in this project is to automatically detect the head gestures in the whole video. Since the number of head gestures are high and manual detection for the whole video is hard and time consuming, automatic detection is preferred. The automatically detected head gestures are used in the gesture-speech correlation analysis. The whole training set for the head gestures is formed from the first 20 minutes of the 25 minute video by manually snipping 14 down gestures and 16 tilt gestures. The features used in the training set are the 2D coordinates of the center of the head and 8 global motion parameters, extracted using optical flow vectors for each frame of the gesture sequence.

We have trained a left-to-right Hidden Markov Model for each gesture and applied an isolated gesture recognition. For measuring the performance, we used only 10 down gestures and 11 tilt gestures and reserved 4 down gestures and 5 tilt gestures for testing. For a given hand trajectory, each gesture model is tested and the one with the maximum likelihood is selected. The average recognition accuracy is given in Table IV. Results are given for HMMs with 5 states and with 5 mixtures of Gaussian.

Gesture	Accuracy on Training Set	Accuracy on Test Set
down	0.883	0.875
tilt	1	1

TABLE IV
ISOLATED HEAD GESTURE RECOGNITION PERFORMANCE

To automatically detect the gestures in a video stream, a gesture spotting methodology must be used. In speech, for keyword spotting, a garbage model is formed as well as the keyword models. The garbage model for speech can be trained with clearly defined non-keywords. However, in gesture spotting, it is not clear what a non-gesture is. To overcome this problem, Lee and Kim [17] proposed a threshold model that utilizes the internal segmentation property of the HMM. For gesture spotting, we have used the approach of threshold model of Lee and Kim. The threshold model is formed by using the states of the gesture models. All outgoing transitions of states are removed and all the states are fully connected such that in the new model, each state can reach all other states in a single transition. Prior probabilities, observation probabilities and self-transition probabilities of each state remain the same, and probabilities of outgoing transitions are equally re-assigned. For a particular sequence to be recognized as a gesture, its likelihood should exceed that of the other gesture models and the threshold model.

Once the threshold model is formed, gesture spotting can be performed. When continuous stream is given as input, we start with the first frame and increase the sequence length

until we identified the sequence as a gesture. After a gesture is identified, spotting continues with the next frame after the end of the gesture. To speed up the process, minimum and maximum lengths for a gesture can be used.

We applied gesture spotting on the 2 minute data. Figure 13 shows the correlation between head gestures and accents in speech. Also the overlap between the manually labelled and automatically spotted head gestures can be seen. When compared to manual labelling, automatic spotting finds *down* gestures more frequently and *tilt* gestures less frequently.



Fig. 13. Correlation between accents and head gestures. Accents are manually labelled. Gestures are both manually labelled and automatically spotted

C. Automatic Recognition of Hand Gestures

In this section we present a framework for hand gesture recognition. Proposed framework consists of three main tasks which are tracking of hand region, extraction of head gesture features and recognition of head gestures based on Hidden Markov Models (HMM). Center of mass position and velocity of each hand is tracked and smoothed using two different filters which are Kalman and Particle Filter. Smoothed position and velocity of center of mass is defined as hand gesture features. HMM is applied for recognition of gestures given the gesture features.

1) *Initialization of Hand Tracker*: In order to extract initial position of hand regions one should exhaustively search for hand in the initial frame. We propose two methods for determination of hand regions in the initial frames.

First methodology is based on skin color information. Given an initial frame, skin colored regions are extracted using the methodology described in Section VI-B.2. However, in this case, unlike the head region correction algorithm, region of interest is selected as whole image. Thus the skin regions coming from head are also marked as candidate hand regions. After obtaining all hand region candidates, connected component analysis is applied to determine connected regions in the set of hand region candidate pixels. The connected components that are larger than or smaller than thresholds t_h and t_s are discarded. In our experiments t_{high} and t_{low} are selected as 100 and 300 pixels. Remaining skin colors are detected as hand region using a semi-automatic method in which software asks user for verification of hand regions.

Second methodology is based on boosted Haar based cascade classifiers. In addition to face detection task VI-B.1 Boosted Haar based cascade classifiers can also be applied to detect hand regions. However, unlike face detection, there is no common classifier structure for hand detection task. Therefore, hand detector classifier is trained using 240 sample hand

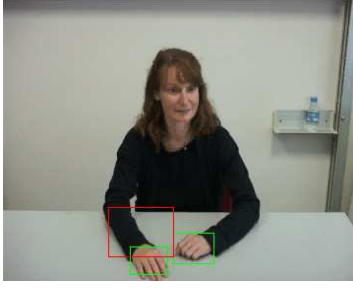


Fig. 14. Initial Hand Regions

regions which are selected from "our database". Recognition performance of Haar based classifier is high however false alarm rate is not as good as Haar based classifier for faces. As an expected result, there is a tradeoff between recognition performance and false alarm rate. In order to detect most of the poses correctly one should train the system using the training samples that contains significant number of poses. This results in an increase in the false accept rate. In contrast if one trains the classifier with specific pose only, the recognition performance of the classifier drops significantly. Therefore, detection and false alarm rate for Haar based classifiers will be high in the task of recognition of hands in specific pose like sign language. In order to decrease the false alarm rate, skin color information is fused with Haar based classifier decisions. Fusion rule is looking at the hand region candidates and check whether there is significant amount of skin colored pixels in candidate region. The threshold t_h for determination of hand region is 10% where the Haar based classifier decision is a bounding box for hand. Sample initial hand positions found by boosted Haar based cascade classifier can be seen as green and red rectangles in Figure 14 where red rectangle is eliminated by using the method described above.

2) *State-Space Model for Kalman Filtering*: Kalman filter based state-space estimator assumes that the motion of a pixel can be approximated using the motion model

$$\begin{aligned} x_t &= x_{t-1} + v_{x,t-1}T + a_{x,t-1}T^2/2 \\ y_t &= y_{t-1} + v_{y,t-1}T + a_{y,t-1}T^2/2 \\ v_{x,t-1} &= v_{x,t-1} + a_{x,t-1}T \\ v_{y,t-1} &= v_{y,t-1} + a_{y,t-1}T \end{aligned} \quad (7)$$

Here, T denotes the frame capture rate of acquisition system which is 1/25 fps. Velocity and acceleration in each direction x, y is represented with v_x, v_y and a_x, a_y respectively. Moreover, motion model defined in 7 can further be approximated by ignoring acceleration term since we can neglect the change in acceleration between two consecutive frames. Neglecting the acceleration has another advantage that each derivative operation is corrupted by noise and simply discarding the higher order derivatives yield better system performance if we consider noise/precision tradeoff. Thus the state motion model becomes

$$\begin{aligned} x_t &= x_{t-1} + v_{x,t-1}T \\ y_t &= y_{t-1} + v_{y,t-1}T \end{aligned} \quad (8)$$

Let the position of center of mass of hand is x, y . The motion of each pixel in two dimensions can be approximated using model 8:

$$x_t = x_{t-1} + v_{x,t-1}T \quad (9)$$

$$y_t = y_{t-1} + v_{y,t-1}T \quad (10)$$

Thus the motion model motivates the following state-space model with state s and observations z :

$$\begin{aligned} s_{t+1} &= Fs_t + Gu_t \\ z_t &= Hs_t + v_t \end{aligned} \quad (11)$$

$$s_t = [x_t, y_t, v_{x,t}, v_{y,t}]^T \quad (12)$$

$$z_t = [x_t, y_t]^T \quad (13)$$

$$F = \begin{bmatrix} 1 & 0 & T & 0 \\ 0 & 1 & 0 & T \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad G = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (14)$$

$$H = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \quad (15)$$

Process noise u_t and the measurement noise v_t are assumed to be uncorrelated, with zero-mean white gaussian distributions and corresponding covariance matrices Q and R . Q and R matrices are adjusted by using method [16] which tests the whiteness of the innovations process.

3) *Hand Tracking with Kalman Filter*: After obtaining the initial hand regions for each hand, we define two search windows based on the bounding box dimensions provided by the hand detector. Two independent Kalman filters are initialized using the positions provided by the hand detector. For each iteration, Kalman filter time update equations are calculated to predict the new hand position. The predicted hand position is used to warp search window. Mean of skin color pixel positions inside the search window is calculated and provided to Kalman filter as the observation. Kalman filter measurement update equations are calculated to correct using the observations. Posterior states of each Kalman filter is defined as feature vectors.

4) *Recognition of Hand Gestures*: The hand gestures that are identified are the gestures performed when the keywords are spoken, so there are three hand gestures related with the three keywords: *left*, *right* and *straight*. The purpose is to build good models of each hand gesture so that the models can be used for the animation part. Recognition performance is used to check the quality of the produced model. The whole training set for the hand gestures is formed from the first 20 minutes of the 25 minute video by snipping the parts that corresponds to manually labelled keywords. We manually checked and eliminated some videos where there is no meaningful gesture even there is a keyword. The final training set contains 20 left, 24 right and 28 straight gestures. The features used in the training set are the 2D coordinates of the center of the left and right hands and their velocities for each frame of the gesture sequence.

We have trained a left-to-right Hidden Markov Model for each gesture and applied an isolated gesture recognition. For measuring the performance, we used 30% of the data for testing. For a given hand trajectory, each gesture model is tested and the one with the maximum likelihood is selected. The average recognition accuracies are given in Tables V, VI, VII. We trained different models for right hand, left hand and using both hands. Each HMM is trained with 5 states and with 1 Gaussian mixture. Although the recognition rates on

Gesture	Accuracy on Training Set	Accuracy on Test Set
left	0.85	0.66
right	1	1
straight	0.8	0.25

TABLE V

ISOLATED HAND GESTURE RECOGNITION PERFORMANCE – USING ONLY
RIGHT HAND

Gesture	Accuracy on Training Set	Accuracy on Test Set
left	0.92	0.7
right	0.76	0.2
straight	0.8	0.5

TABLE VI

ISOLATED HAND GESTURE RECOGNITION PERFORMANCE – USING ONLY
LEFT HAND

Gesture	Accuracy on Training Set	Accuracy on Test Set
left	1	0.83
right	1	0.71
straight	1	0.70

TABLE VII

ISOLATED HAND GESTURE RECOGNITION PERFORMANCE – USING BOTH
HANDS

test set are low, training set accuracies are high. This result indicates that the generalization ability of the learned models are poor but the models are good at creating the training data. Therefore, these HMM models can be used for animation purposes rather than recognition purposes. The technique used for restoring the hand trajectory using the produced models are given in detail in the Animation section.

VII. ANIMATION

A. Stick Model and 3D Body Model

Given a speech sequence, keyword spotter described in Section V-B.1 and accent detector described in Section V-B.2 are used to extract time-stamps of auditory events. These time-stamps and speech sequence are provided to animation engine to animate the virtual body. Initially virtual body is at the stable state and for each frame, animation engine checks for the time-stamps. If there is a coincidence between time-stamps and frame-stamps, corresponding body part is actuated to be

at a moving state. In this project, we realized two animation schemes:

1) *Stick Model*: Stick Model consists of line segments that corresponds to forearm and upper arm where starting and ending points of these line segments are determined as hand, shoulder and elbow positions. Together with these line segments head is included with a line segment between head position and the center of the line segment between left and right shoulder. Animation engine for Stick Model uses 2D coordinates of the corresponding points.

2) *3D Body Model*: 3D Body Model consists of 2 arms and head without the body. Animation engine for this model uses a dictionary of gestural events and frames are constructed manually for each event in the dictionary. Animation engine uses each event independently for the animation of head, left arm and right arm.

B. Head and Hand Motion Models

In order to animate the body model, the center of mass positions of head and both hands is required by the animation engine. For each acoustical event, related gesture synthesized by considering the duration of acoustical event and the previously recognized gestures.

1) *Hand Motion Model*: Figure 15 shows the trajectories of left and right hand for the hand gesture examples in the training set. Each trajectory is shifted such that the origin is (0,0). For the left gesture, the motion of the right hand is limited when compared to the motion of the left hand. Similarly for the right gesture, the motion of the left hand is limited when compared to the motion of the right hand. However for the straight gesture, both hands have large trajectories. The hand models for each hand gesture are constructed by HMMs. For the left gesture, we trained an HMM by using only the left hand trajectory, for the right gesture, we trained an HMM by using only right hand trajectory and for the straight gesture we trained two HMMs: one for the left hand and one for the right hand.

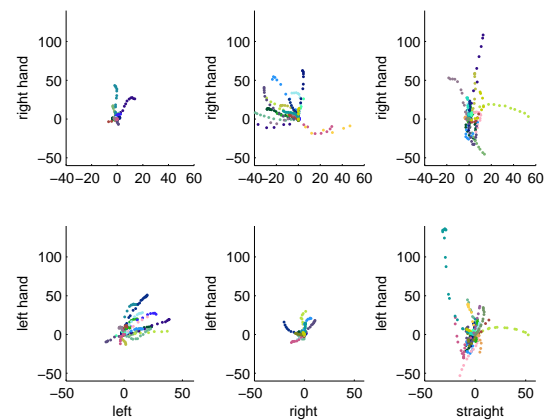


Fig. 15. Left and right hand trajectories of the gestures

To construct an observation sequence from the HMM models, we used the model parameters: state transaction probabilities, parameters of gaussian distribution (covariance matrix

and mean for the feature vector) for each state and prior probabilities of states (since HMMs are left to right, always start with the first state). Using these information, we can construct an observation sequence by just providing a sequence length. To have an idea about the sequence lengths (number of frames) of the hand gestures, we first draw the histogram of sequence lengths and then applied a normality test. Figure 16 shows the histograms and plots of the normality tests for each gesture. The test results show that the distribution of sequence lengths for each gesture is close to normal distribution. If there are no other information about the sequence length when constructing an observation sequence, a random length can be selected from the related distribution. Table VIII shows the normal distribution parameters for gesture types.

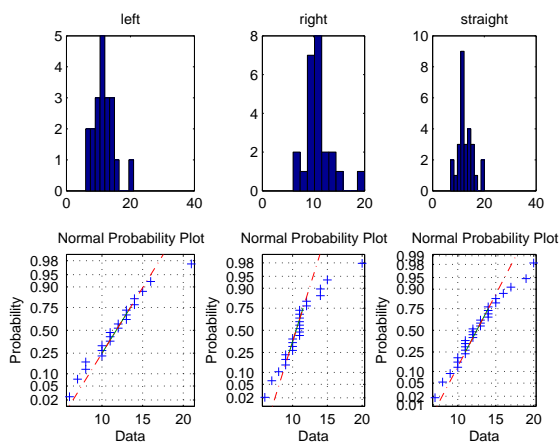


Fig. 16. Sequence length histograms and normality test plots

Gesture	μ	σ
left	11.75	3.43
right	10.96	2.82
straight	12.64	3.02

TABLE VIII

NORMAL DISTRIBUTION PARAMETERS FOR THE GESTURE LENGTHS

The methodology used for constructing the observation sequence, given a sequence length and model parameters is as follows:

- 1) Generate random numbers between [0,1) for each observation (so the number of random numbers generated must be equal to the sequence length)
- 2) Using the prior probabilities and the first random number, decide the first state (e.g. If there are 3 states with prior probabilities 0.2, 0.5, 0.3, then the decision is given by first taking a cumulative sum of the probabilities, which is 0.2, 0.7, 1. If the generated random number is less than 0.2, then select state 1 as the first state. If the random number is between 0.2 and 0.7, select the second state and if it is higher than 0.7, select the third state.
- 3) For $i=2$:sequence length

- 4) Decide i^{th} state using the state transition probabilities, the previous state and the i^{th} random number
- 5) End for
- 6) For $i=1$:sequence length
- 7) Construct the i^{th} observation by producing a random number from the gaussian distribution of the i^{th} state
- 8) End for

By using this methodology, we produced hand trajectories for each gesture where, for the *left* gesture, only left hand moves; for the *right* gesture, only right hand moves; and for the *straight* gesture both hands move.

On the last 5 minutes of the database, we first run the keyword spotting algorithm for finding the time-stamps for words *left*, *right* and *straight*. We then produced the related hand gestures which are animated during the same period with the keyword. The sequence length is determined by the length of the keyword. Figure 17 shows the produced trajectories. As seen from the figure, left gestures are aiming left and right gestures are aiming right and straight gestures move in the y direction. This plot is similar to the one of the training data (Figure 15).

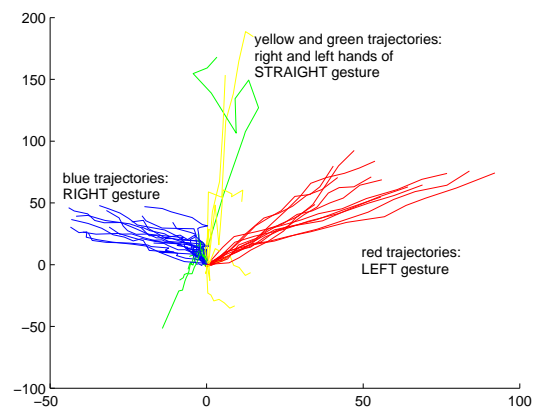


Fig. 17. Hand trajectories of the produced gestures

2) *Head Motion Model*: Head motion model is generated according to the duration of accents. Let the duration of the accent be t_a seconds. For $t_a/2$ seconds head center of mass is shifted in $+y$ direction with 25 pixels/second. For the remaining $t_a/2$ seconds head center of mass is shifted back to its resting positions. Practical aspect of this methodology is that, the accents with short period are visually eliminated and the accents with long period are visually amplified.

VIII. CONCLUDING REMARKS AND FUTURE WORK

In this project, gesture synthesizer based an audio-visual correlation is presented. Audio-visual correlation analysis is conducted using acoustic and visual events. Acoustic events are divided into semantic and prosodic categories. Visual events are selected as hand and head gestures. The types of events are defined by investigating a portion of the database. The repetitive patterns for acoustic events are mainly keywords (*left*, *right* and *straight*) and accents. The repetitive patterns for head gestures are *nod* and *tilt*. *Left* movement of left hand,

right movement of right hand and *down* movement of both hands are defined as hand gestures.

Given a limited number of examples for each kind of event, an event model is created. Using the training portion of the database, these events are spotted and co-occurring patterns are investigated to train the correlation model. Spotting of keyword, accents and head movements was not problematic and spotting performances were high enough to be applicable. However hand movement spotting rate was not high enough since the hand movements for gestures are not well defined motions.

Investigating the co-occurring patterns, we concluded that keywords and corresponding hand movements are strongly correlated. Moreover, *nod* movement of head is found out to be highly correlated with accents. Motivated from this fact, using the test portion of the database, first, keywords and accents are detected. Then the virtual body is animated using corresponding visual event at those detected acoustic events.

As a future work, our ultimate goal is building up new audio-visual databases. The scenario of the database that is used in our project is "Direction Giving" and the scenarios can also be extended in new databases. The number of keywords and gesture patterns will be increased using new scenarios for synthesis of more natural gestures.

ACKNOWLEDGMENT

Authors would like thank Ana Huerta Carrillo and Arman Savran for their help in realizing the 3D Animation. We also thank Hannes Pirker for inspiring discussions.

REFERENCES

- [1] Y. Yasinnik, M. Renwick, S. Shattuck-Hufnagel, "The Timing of Speech-Accompanying Gestures with Respect to Prosody," Conf. of The From Sound to Sense, 2004.
- [2] S. Duncan, F. Parrill, D. Loehr, "Discourse factors in gesture and speech prosody," Conf. of the International Society for Gesture Studies (ISGS), Lyon, France, 2005.
- [3] Jie Yao and Jeremy R. Cooperstock, "Arm Gesture Detection in a Classroom Environment," Proc. WACV'02 pp. 153-157, 2002.
- [4] Y. Azoz, L. Devi. R. Sharma, "Tracking Hand Dynamics in Unconstrained Environments," Proc. Int. Conference on Automatic Face and Gesture Recognition'98 pp. 274-279, 1998.
- [5] S. Malassiotis, N. Aifanti, M.G. Strintzis, "A Gesture Recognition System Using 3D Data," Proc. Int. Symposium on 3D Data Processing Visualization and Transmission'02 pp. 190-193, 2002.
- [6] J-M. Chung, N. Ohnishi, "Cue Circles: Image Feature for Measuring 3-D Motion of Articulated Objects Using Sequential Image Pair," Proc. Int. Conference on Automatic Face and Gesture Recognition'98 pp. 474-479, 1998.
- [7] S. Kettebekov, M. Yeasin, R. Sharma, "Prosody based co-analysis for continuous recognition of coverbal gestures," Proc. ICMI'02 pp. 161-166, 2002.
- [8] F. Quek, D. McNeill, R. Ansari, X-F. Ma, R. Bryll, S. Duncan, K.E. McCullough "Gesture cues for conversational interaction in monocular video," Proc. Int. Workshop on Recognition, Analysis, and Tracking of Faces and Gestures in Real-Time Systems'99 pp. 119-126, 1999.
- [9] P. Boersma, "Accurate short-term analysis of the fundamental frequency and the harmonics-to-noise ratio of a sampled sound," Proc. of the Inst. of Phonetic Sciences 17: pp. 97-110, 1993.
- [10] A. L. Ronzhin, Y. A. Kosarev, I.V. Lee, A. A. Karpov, "The method of continuous speech recognition based on signal analysis in a sliding window and theory of the fuzzy sets". In scientific-theoretical journal "Artificial intelligence", Donetsk, Ukraine, 2004. vol. 4. pp. 256-263.
- [11] For detailed information visit: <http://htk.eng.cam.ac.uk>

- [12] J. Caminero, C. de la Torre, L. Villarrubia, C. Martin, L. Hernandez, "On-line garbage modeling with discriminant analysis for utterance verification". In Proc. of 4-th International Conference on Spoken Language Processing ICSLP 96, Philadelphia, PA, USA, pp. 2111-2114.
- [13] P. Viola and M.J. Jones, "Rapid Object Detection using a Boosted Cascade of Simple Features", IEEE CVPR, 2001.
- [14] R. Lienhart and J. Maydt, "An Extended Set of Haar-like Features for Rapid Object Detection", IEEE ICIP 2002, Vol. 1, pp. 900-903, Sep. 2002.
- [15] Jean-Yves Bouguet, "Pyramidal Implementation of the Lucas Kanade Feature Tracker Description of the algorithm", Intel Corporation, Microprocessor Research Labs, OpenCVD Documents, 1999
- [16] R. K. Mehra, "On the identification of variances and adaptive Kalman filtering", IEEE Transactions on Automatic Control, AC-15, pp. 175-183, 1970.
- [17] H. Lee and J. H. Kim, "An HMM-Based Threshold Model Approach for Gesture Recognition", IEEE Transactions on Pattern Analysis Machine Intelligence, 21, 10 (Oct. 1999), 961-973.
- [18] L. R. Rabiner, "A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition", Proceedings of the IEEE, Vol. 77, No. 2 (February. 1989).
- [19] J. Hirschberg and G. Ward, "The influence of pitch range, duration, amplitude and spectral features on the interpretation of the rise-fall-rise intonation contour in English", Journal of Phonetics, v.20, n.2, pp. 241-251, 1992.
- [20] Epstein, M.A., "Voice quality and prosody in English. Proceedings of the XVth International Congress of Phonetic Sciences", ICPhS 03, 2003.

APPENDIX I SOFTWARE NOTES

A. Codes related to head and hand gesture recognition and hand gesture modeling

For HMM training HMM routines in "Bayes Net Toolbox for Matlab, written by Kevin Patrick Murphy et al." are used. To run the codes, this BNT toolbox must be in the path of Matlab.

For head gesture related routines, check the folder head - run `readHeadavi.m` or `readHeaddata.m` or `readHead2min.m` to read and save data in a mat file OR write your own routine to read data. You may have to change the paths for some files - to train HMMs with continuous observations run `trainHMMsCont`, but first modify the name of mat file if needed - `gesturespot` routine runs for a sequence and spots the gestures and non gestures

For hand gesture related routines, check the folder hand - run `readHanddata.m` to read and save data in a mat file - `trainHMMsCont` trains HMM models for each gesture - `traindiffHMMsCont` trains different HMM models (using different length of feature vectors) for each hand gesture - `plottrajectory` plots the trajectories of gestures in the training set - `plotartificialtrajectoryCont` plots produced trajectories - `generatetesttraj` generates trajectories acc to the timestamps of keywords in speech

Multimodal Caricatural Mirror

Martin O.⁽¹⁾, Adell J.⁽²⁾, Huerta A.⁽³⁾, Kotsia I.⁽⁴⁾, Savran A.⁽⁵⁾, Sebbe R.⁽⁶⁾

(1) : Université catholique de Louvain, Belgium

(2) Universitat Polytechnica de Barcelona, Spain

(3) Universidad Polytécnica de Madrid, Spain

(4) Aristotle University of Thessaloniki, Greece

(5) Bogazici University, Turkey

(6) Faculté Polytechnique de Mons, Belgium

Abstract—This project aims at creating a multimodal ‘caricatural’ mirror, where users see and hear their own emotions amplified by an avatar, mimicking the user’s facial expressions and prosody using a wide screen and loudspeakers. The goal of the project is also to bring together researchers from various fields so as to build a whole system using everyone’s expertise. The main technical challenges include facial animation, automatic face tracking, automatic vocal and facial features extraction and multimodal emotion recognition and synthesis.

Index Terms—Emotion Recognition, Face analysis, Prosody analysis, Facial Animation, Prosody synthesis, Multimodal Interfaces

I. INTRODUCTION

Researches on automatic emotion recognition and synthesis is currently focusing the attention of an ever-growing community of researchers from various fields (signal processing, artificial intelligence, psychologists, human-computer interactions, ...). Many different prototypes of emotion recognition systems have already been developed but it remains very difficult to compare the results of such systems, due to the lack of common databases and experimenting protocols.

In this project, we will then focus on the development of a system that would involve the user both interactively and emotionally, giving the opportunity to users to assess the usability of such a system, while at the same time generating real emotional experiences on the user’s side. This affective response from the user will be used to further train the system, the emotions generated being expressed in a more natural way than when expressed ‘on demand’, as it is the case for most of the existing databases. We could then view the system as a way of building a multimodal database, which could later be used by the researchers of the SIMILAR network of excellence to compare the performances of their emotion recognition algorithms.

This paper will be divided in three major sections. The first section describes the visual modality, which involves both the analysis and synthesis of facial expressions. The second section will deal with the automatic analysis of user’s prosody and the exaggeration of the extracted prosodic features, in order to ‘caricaturate’ the user’s voice. Eventually, the last section concerns the integration of both modalities for synchronized and realistic multimodal facial animation.

II. VISUAL MODALITY

In this section, we will describe the technical challenges related to automatic analysis and synthesis of facial expressions. In the first time, we will describe the steps that have to be achieved to capture the user’s emotional state. We will finish our description of the visual modality by presenting the techniques involved in the generation of realistic facial animation.

When tackling the problem of automatic analysis of facial expressions, one generally decomposes it in three sub-problems:

- Automatic Face Detection and Tracking
- Automatic Facial Features Detection and Tracking
- Automatic Expression Recognition/Classification

We will then successively detail each of these challenges, along with the solutions we implemented.

A. Face Detection

The first thing to do when one wants to design a facial expression recognition system is to select the experimental conditions under which experiments will have to be run. In our case, as we want the system to be fully automatic, we have to start by detecting the user’s face inside the scene.

Although it seems like an easy problem at first glance, we quickly realized that the high variability in the types of faces

encountered makes the automatic detection of the face a tricky problem. After discussing within the team the techniques that could be used for face detection and tracking, we searched in the literature for existing prototypes.

Many different techniques have been tried in order to solve the problem of detecting a face in a scene. After a deep inspection of the state-of-the-art, it appears that there isn't a unique solution to the problem. Rather, the best face trackers have been obtained by using a combination of the available techniques. A technique formally known as *boosting* seems to suit particularly well our needs: the joint use of several weak detectors (classifiers which are not able to precisely detect a face in a scene) may lead to a robust face detector. The robustness is achieved by exploiting the independence between the criteria used by the different individual detectors. To understand how the *boosting* is achieved, we need to detail a bit the construction of our face detector. First, each of the individual detectors D are trained over the entire training set, producing a recognition rate R_D . Then, we assign a score S_D to each of the D classifier, according to its recognition rate R_D . The face detector then balance the influence of each of the individual detectors on the final decision, according to their relative individual performances.

In the scope of this project, we chose to use an open-source implementation of such a boosted face tracker. After comparing existing systems, we decided to use the OpenCV [1] face tracker. Already trained on a large database of face/non-face images, it produces efficient face detection in all kinds of settings, thus completely fitting our needs. The result obtained with this face tracker is depicted in the figure below.



Figure 1 : Output of the OpenCV face tracker

In practice, the face tracker finds the face localization on the first image of the video sequence and surrounds it with a bounding box. The resulting face image is then passed to the facial features detection module whose goal is to initialize the facial features tracker.

B. Facial Features Detection

The main goal of the facial feature detection algorithm is to specify the exact position for a set of desired points on the image. These points will be later used for the extraction of the facial features position. As crucial points are considered the points that correspond to the inner and outer side of the eyebrows and eyes, as well as the corners of the mouth, thus forming a set of 10 points in total.

The method proposed by Sobottka et al. [2], was followed to extract the position of the desired points in the image. Facial features consist of a facial region of particular interest, as they differ from the rest of the face due to their low brightness. Therefore, an analysis of the local minima of the brightness values can specify the correct position for the facial features that are included in the region under examination.

The image is converted in black and white format, to make the whole procedure more robust. Let $\{x,y\}$ be the coordinates of each facial feature under examination. The image is divided in columns, forming in that way three equal regions. The interval in which the y coordinate of the eyes region belongs, can be obtained by examining the pixels' brightness at the columns that correspond to the one third and two thirds of the image's width (left and right eye region respectively). Following the same method, the interval in which the y coordinate of the mouth region belongs, can be obtained by examining the column that corresponds to the middle of the image's width.

The graphic plot of the pixels' brightness at those specific columns indicates the position of local minima in the image. Those local minima correspond to the eyebrows and eyes region for the case of the column at the 1/3 and 2/3 of the image's width, and to the mouth area for the case of the 1/2 of the image width. The local minima actually appear in a sequential way, just like the way the facial features appear in a facial image if that is analyzed from the top to the bottom.

At the images below, someone can actually see the way the eyebrows, eyes and mouth regions are defined by the lower values of their brightness.

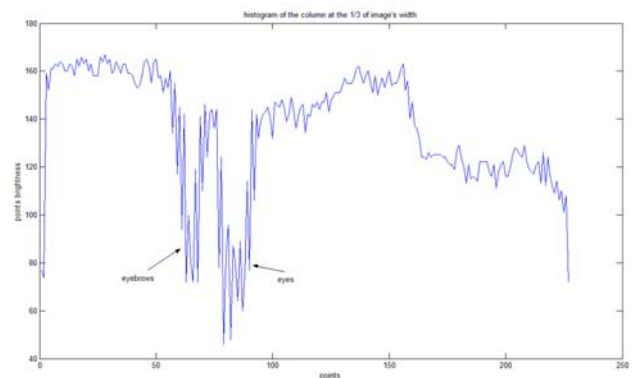


Figure 2: Pixels brightness at the column corresponding to the 1/3 of the image width

The same procedure is followed to define the x coordinates of the facial features region in the image, with pixel brightness values taken now for specific rows ($1/3$ of the image's height for the eyes region and $2/3$ of the image's height for the mouth region).

In that way, three big regions are defined, one for each eye as well as the mouth (see Figure 5 below).

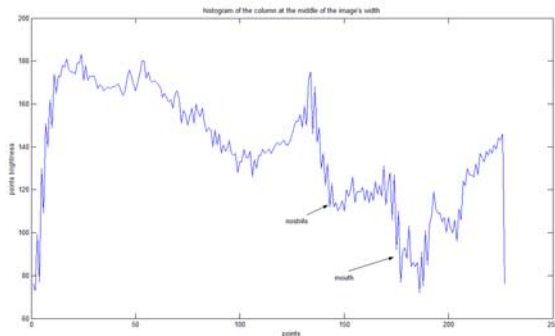


Figure 3: Pixels brightness at the column corresponding to the $1/2$ of the image width

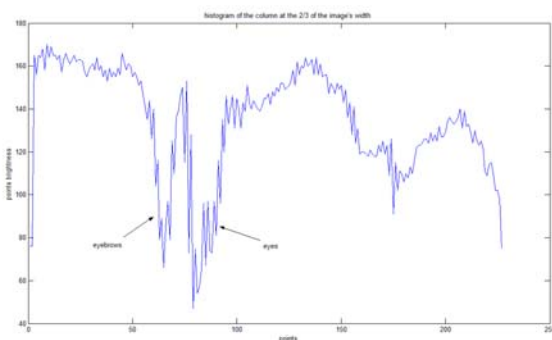


Figure 4: Pixels brightness at the column corresponding to the $2/3$ of the image width

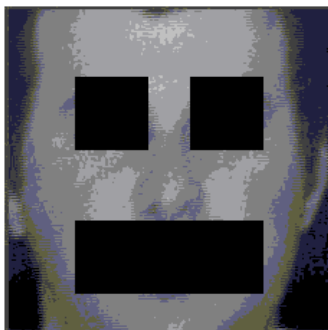


Figure 5: Facial features regions detected in the first frame of the image sequence



Figure 6: Facial features detected in the first frame of the image sequence

In the above-mentioned regions, every desired facial feature is uniquely characterized. For example, the outer corner of the left eyebrow is the first dark pixel at the top up and left corner, while the inner corner of the eyebrow is the first dark pixel at the top up and right corner. The eyes are more difficult to detect, but if the eyebrow region is defined, the follow similar rules apply for the region left unused. That way all of the desired facial features are detected, so as to be used for the initialization of the Candide grid to the first frame of the image sequence, as described in the following section (Figure 6).

C. Facial Features Real-Time Tracking

The facial features information extraction is performed by a grid adaptation system, based on deformable grid models.

The algorithm is based on tracking a large number of previously selected feature points in the facial region as depicted at the first frame of the video sequence. The video sequence progresses in time, depicting the facial expression evolving, to reach its highest intensity at the last frame of the video sequence. The nodes of the fitted Candide grid (output of initialization process) are tracked using a pyramidal implementation of the well-known Kanade-Lucas-Tomasi (KLT) algorithm [3]. As soon as the tracking algorithm computes the displacement of all the tracked features, the resulting configuration (containing the new positions of the model nodes) is deformed. The displacements of model nodes, are assumed to be the driving forces of the model deformation, thereby providing an accurate and robust model based facial feature tracking method.

The system can be seen in Figure 7.

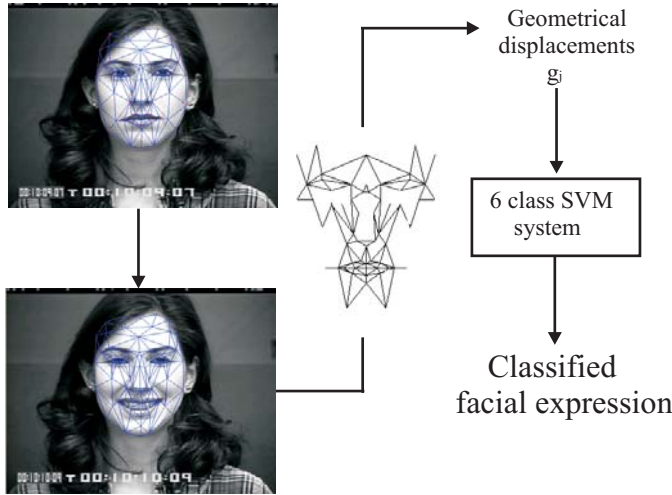


Figure 7: Diagram flow of the system used for facial expression classification

The Candide grid is automatically initialized on the first frame of the image sequence, depicting a face in its neutral state. To achieve that, the points detected following the procedure described at the previous section are used. The points chosen for the initialization were those of the greatest importance, being the ones responsible for the formation of facial movement according to *Facial Action Coding System (FACS)*[4]. The software automatically adjusts the grid to the face and then tracks it through the image sequence, following the facial expression landmarks evolving through time [5]. At the end, the grid adaptation software produces the deformed Candide grid that corresponds to the full facial expression appearing at the image sequence.

The deformed Candide grid produced by the grid adaptation algorithm [5], that corresponds to the greatest intensity of the facial expression shown, contains 104 nodes. Only some of these nodes are important for the recognition of the facial expressions. For example, nodes on the outer contour of the face do not contribute much to facial expressions. Thus, a subset of 62 nodes is chosen, that control the movement according to *FACS* used for describing the facial expressions, so as to perform facial expression recognition [6].

D. Facial Expression Recognition using Support Vector Machines

The classification is performed based only in geometrical information, not taking into consideration any luminance or color information. The geometrical information used is the displacement of one point d_i , defined as the difference between the last and the first frame's coordinates

$$d_i^c = \begin{bmatrix} \Delta x_i^j(c) \\ \Delta y_i^j(c) \end{bmatrix}, \quad i = 1, \dots, 62 \text{ and } j = 1, \dots, n \quad (1)$$

where c is the number of classes of facial expressions to be recognized, here equal to 6, i is the number of points taken under consideration, here equal to 62 and j is the number of image sequences to be examined.

In that way, for every image sequence to be examined, a feature vector F_j^c is constructed, containing the geometrical displacement of every point taken into consideration, thus having the following form

$$F_j^c = \begin{bmatrix} d_1^j(c) \\ d_2^j(c) \\ \dots \\ d_{62}^j(c) \end{bmatrix} \quad (2)$$

That feature vector is used as an input to a multi class Support Vector Machine system (SVM), with six classes implemented for the experiments, that classifies each set of grid's geometrical displacements to one of the 6 basic facial expressions (anger, disgust, fear, happiness, sadness and surprise).

Let $S = \{ \{F_j^c\}_{j=1}^n, l_j \}$ the training data set of labeled training patterns, $F_j \in R^d$, where n is equal to the number of grids to be examined, d denotes the dimensionality of the training patterns, here equal to $62 \times 2 = 124$, and $l_j \in \{1, \dots, 6\}$.

The decision function $f(F, a)$, which classifies a vector F , is chosen from a set of functions defined by the parameter a . The parameter a should be chosen in such a way that for any F the function should be able to provide a classification l as close to the estimation as possible.

The main idea of an SVM system is to construct a hyperplane that will separate the desired classes, in such a way that the margin (defined as the distance between the hyperplane and the nearest point) is maximal. Therefore, to generalize, the following equation should be minimized

$$\phi(w, \xi) = \frac{1}{2} \sum_{m=1}^6 (w_m \cdot w_m) + C \sum_{i=1}^n \sum_{m \neq y_i} \xi_i^m \quad (3)$$

with constraints

$$(w_{l_i} \cdot F_i) + b_{l_i} \geq (w_m \cdot F_i) + b_m + 2 - \xi_i^m \quad (4)$$

$$\xi_i^m \geq 0, i = 1, \dots, n \quad m \in \{1, \dots, 6\} \setminus l_i \quad (5)$$

The decision function that is derived from equations 3 and 4 is the following

$$f(F) = \arg \max_k [(w_i \cdot F) + b_i], i = 1, \dots, 6 \quad (6)$$

The solution to this optimization problem in dual variables can be found by the saddle point of the Lagrangian

$$L(w, b, \xi, \alpha, \beta) = \frac{1}{2} \sum_{m=1}^6 (w_m \cdot w_m) + C \sum_{i=1}^6 \sum_{m=1}^6 \xi_i^m - \sum_{i=1}^6 \sum_{m=1}^6 \alpha_i^m [((w_{l_i} - w_m) \cdot F_i) + b_{l_i} - b_m - 2 + \xi_i^m] - \sum_{i=1}^6 \sum_{m=1}^6 \beta_i^m \xi_i^m \quad (7)$$

with the dummy variables

$$\alpha_i^{l_i} = 0, \xi_i^{l_i} = 2, \beta_i^{l_i} = 0, i = 1, \dots, 6 \quad (8)$$

and constraints

$$\alpha_i^m \geq 0, \beta_i^m \geq 0, \xi_i^m \geq 0, i = 1, \dots, 6 \quad m \in \{1, \dots, 6\} \setminus l_i \quad (9)$$

which has to be maximized with respect to α and β and be minimized with respect to w and ξ .

By further processing [7] equation 6 is finally expressed as

$$f(F, a) = \arg \max_m \left[\sum_{i=1}^m (c_i^m A_i - a_i^m) (F_i \cdot F) + b_m \right] \quad (10)$$

or equivalently

$$f(F, a) = \arg \max_m \left[\sum_{i: l_i = m} A_i (F_i \cdot F) - \sum_{i: l_i \neq m} a_i^m (F_i \cdot F) + b_m \right] \quad (11)$$

where a is a parameter that defines which function is suitable for correctly classifying a vector F , c_i^n is the following notation

$$c_i^n = \begin{cases} 1 & \text{if } l_i = n \\ 0 & \text{if } l_i \neq n \end{cases} \quad (12)$$

and A_i is defined as

$$A_i = \sum_{m=1}^6 \alpha_i^m \quad (13)$$

The SVM system created constructs a maximal linear classifier in a high dimensional feature space, $Z(x)$, defined by a positive kernel function, $k(F, F')$, specifying an inner product in the feature space,

$$Z(F)Z(F') = k(F, F'). \quad (14)$$

The kernel used was a 3rd degree polynomial function, defined in general as

$$k(F, F') = (F \cdot F' + 1)^d \quad (15)$$

where d was equal to 3.

E. Facial Animation

To give the caricatural mirror effect, we decided to generate the visual features of the user on a face model. First of all, the existing open source facial animation softwares were searched, but we could not find a proper one for our task. Therefore, we decided to develop our own facial animation engine with available face models. Candide3 [A2] was the best choice among the other face models, since it includes action units (AUs) and MPEG-4 FAPs. It is a simple 3D mesh model including 184 polygons as shown in Figure 8.

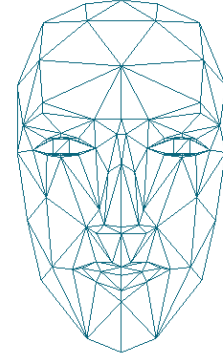


Figure 8 Wireframe model of Candide3

The software for the animation was written using OpenGL, and SDL libraries, since they are open source and run on many platforms. OpenGL is a well-known API for 3D rendering, and SDL provides low-level interfaces to reach media devices like video and sound card. Using these APIs, our software is capable of animating the face and playing the sound synchronously.

There are two different animation procedures in this project: from facial feature tracking and emotion recognition.

1) Animation From Facial Feature Tracking

In this procedure, the tracked facial features are directly used to animate the face model. Since in the tracking algorithm the same face model, Candide3, is employed, the features, which are the coordinates of the vertices of this polygonal model, are used without any operation to deform the polygonal face mesh. However, due to the prosody amplification in the speech, the speech rate is usually altered and therefore a synchronization problem between speech and lip movements occurs. To overcome this, the animation frames are simply scaled in time, and the final animation is obtained.

2) Animation From Emotion Recognition: In this part, for each detected emotion an animation sequence is generated. However, since our task is to generate the emotion not the

speech animation, we decided to ignore the visual speech and made our face model just say "mamama...". By this way, our job for lip synchronization is simplified and we just consider the mouth opening and closure for synchronization.

To generate the emotions, first of all, static expressions were prepared. In our case, there are six emotions as happiness, surprise, sadness, anger, fear and disgust, and a neutral mood. Also, for each emotion we have three states, mouth closure for silence, pressed lips during the sound /m/ and mouth opening for /a/. Modifying the parameters of the Candide3 model created these expressions. There are 65 animation units that control the local movements on the face, like movements for lips and eyebrows, and are realized by simple vertex translations. The prepared expressions are depicted in Figure 9.

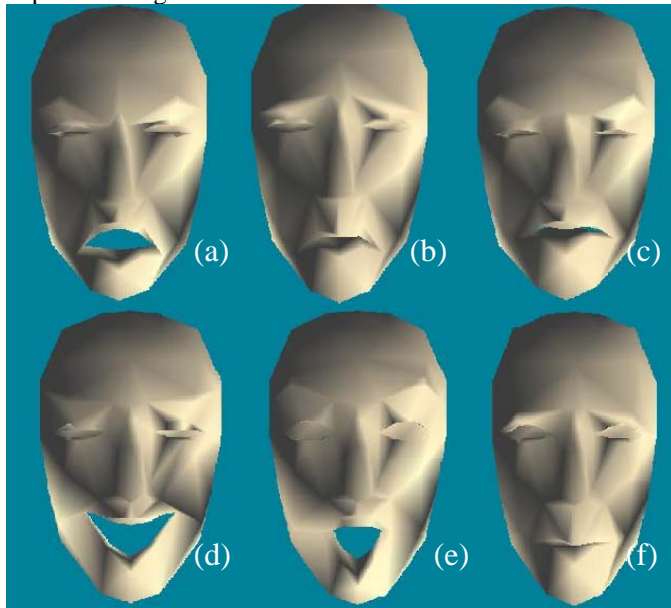


Figure 9 The six basic emotions (anger (a), sadness (b), disgust (c), happiness (d), surprise (e), fear (f)) that were prepared with Candide3

However, there is an important issue in animating a face model. There should be smooth transitions between the expressions otherwise animations seem jerky. For this purpose, sigmoid function (Eq 16), which is a monotonously increasing function, is employed.

$$y = \frac{1}{1 + \exp(-x)} \quad (16)$$

Using the sigmoid function, first, the transitions for the vowel /a/ were modeled. The duration for the sound /a/ is divided into three regions: the entrance phase, steady-state phase and the exit phase. During the steady state, just the target expression is displayed, but for the transitions phase interpolations with the sigmoid function are performed. For the interpolation, a sigmoid function for the target state (/a/) and an inverse of it (1-sigmoid(x)) for the other state were employed to determine the weights for each expression. To model the transitions with sigmoids we need two parameters:

one for the position, and one for the time scaling. The sigmoids are positioned at the center of the transition regions, which are chosen as 45% of the duration for the both entrance and exit phases by observation. By this way, the smoothness of the transition is also automatically adapted to the speaking rate, for example we have sharper transitions with higher speaking rates. On the other hand, with the scaling factor, we have a control on the smoothness. Again empirically the scaling factor is chosen as three. Moreover, we need also smooth transitions for emotions. For this task, again, sigmoid interpolation is performed, by the same procedure as for the /a/ sound.

After obtaining weights for the expressions, the next step is to calculate the weighted sum of the model coordinates. However, instead of directly calculating the vertex coordinates, first, weighted sum of animation unit parameters are computed and then they are applied to the model.

III. VOCAL MODALITY : PROSODY PROCESSING

A. Introduction

The main goal of the application is to be able to emphasize the expression generated by the user in both image and vocal features. In speech it is well known that prosodic parameters carry most of the expressive and emotional information (citation). Due to this, in this project we have focused on the amplification of expressive variation of such parameters. There are three main parameters that are considered to constitute prosody: Fundamental frequency, rythm and energy. We have discarded the energy parameter due to its weakness (citation) and only rythm and pitch have been taken into account. Although there are other parameters of prosody such as voice quality that could support the amplification of expressive events, they have been discarded after preliminar experiments.

When trying to amplify emotional events it is very important to keep the linguistic structure of speech. Speech is a process that carries information in all their aspects and speech parameters are interrelated within each other. The modification of one parameter may influence another one and the linguistic structure contained in speech might be lost. Therefore, since we do not want to distort speech but to emphasize expressive events, it is needed to be done in such a way that language is not disturbed. This idea has been present throughout all the work presented here.

In the system presented here the speech is recorded from the user, some characteristics are extracted from the voice and used to generated models that will lead the speech modification. Afterwards the modification of the original voice is performed and finally played back together with the animation generated. In figure X there is a global overview of the whole process.

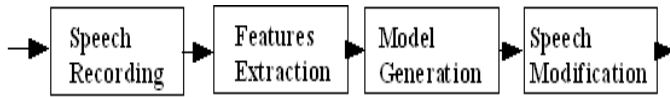


Figure 10:
General overview of the speech transformation module.

B. Prosodic Features Extraction

We planned to modify the pitch and the rhythm of the user speech. These parameters had to be extracted from the original voice. In order to extract them different techniques have been used for each of them. In order to extract the pitch the autocorrelation method that is implemented on PRAAT has been used. This program gives a set of times stamps with a pitch value for each. Then, linear interpolation is performed to fill the unvoiced regions. Finally, we get a contour that has a value for every time stamp. In order to eliminate micro prosody and some effects related with the errors of the algorithm, the contour is smoothed by means of a low pass filter with a cut off frequency of 10Hz.

On the other hand, it is needed to extract the speech rate. In the application presented here the system need to be fast so we could not perform a recognition task to count phonemes it had to be done from the raw signal itself. To tackle this problem some speech exclusive characteristics have been taken into account. As can be seen in the spectrogram in Figure X the speech segments with higher energy are fricatives and vowels, and the difference between them is that fricatives contain their energy mainly in high frequencies.

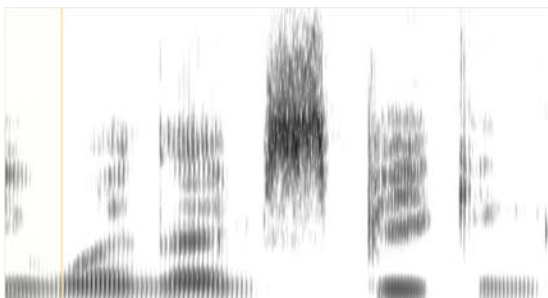


Figure 11:
Spectrogram of "To see" we can see how fricative contain more energy in high frequencies. Above 2KHz.

We can profit this effect by filtering low pass the signal with a cut off frequency of 2KHz. Then energy peak detection is performed. Since major part of the energy is contained in vowels after filtering, these peaks can be considered as an estimate of the speaking rate. It is measures in vowels/second.

In summary, two mean features are extracted: Pitch and Speaking rate. The first one using a classical algorithm in

speech processing and the second one is only estimated by a simple algorithm that tries to find vowels position.

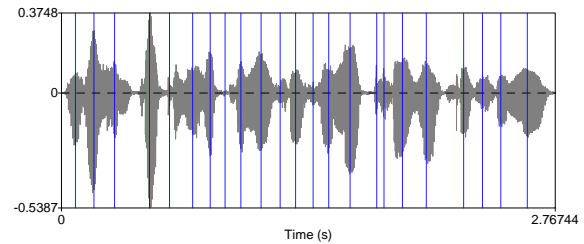


Figure 12:
Results from detecting energy maxima, mainly related to vowels. Then the speech rate is roughly measured as vowels/second.

C. Prosody Amplification

1) Pitch Variations Amplification

Pitch has a drift in a sentence that makes its mean go down. It is due to the fact human lungs are limited and the pressure of the air flowing through the vocal strings diminishes over time. Then, the pitch decreases when this pressure decreases. We have considered that this drift is a base pitch around which variations occur. Then we decided to amplify variations around this drift line. In order to calculate this drift a regression line can be performed when working only with a single sentence. But since in the present project information about beginning and ending sentences is not known, phrase breaks needed to be detected. However, this is a still left to solve problem. Then, in order to avoid it a highly smoothed base pitch line has been used. These base line is a rough approach to calculate the pitch drift but taking into account pitch jumps in phrase brakes.

This base line pitch contour is extracted using the same algorithm as the one described in previous section but with a cut off frequency of 0.5 Hz instead of 10Hz.

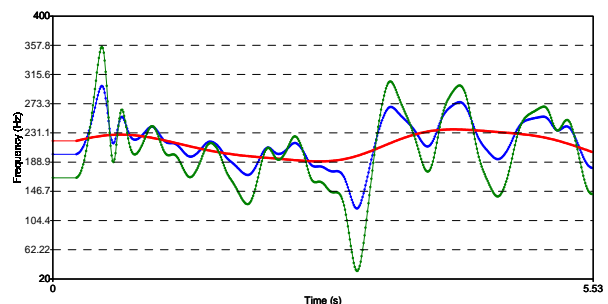


Figure 13:
In a blue line the original contour of the voice is shown. The red line is the base pitch line obtained by smoothing the contour. Green line shows the new generated contour that has been amplified with respect to the base line.

The objective of this base pitch line is to model the natural drift that pitch contour suffers. In order to keep this natural drift the smoothed contour is removed from the original contour. Therefore, the contour is amplified by multiplying it by a parameter. Then accented parts of the pitch are emphasized in a way that large movements are more amplified than small ones as can be seen in Figure X. Let f be the original contour, f_b the base line pitch contour and a a parameter, then the generated contour f_o is:

$$f_o(t) = a * (f(t) - f_b(t)) \quad (17)$$

The resulting speech is more emphasizing since pitch excursions are larger.

2) Speech Rhythm Amplification

When working with speech flow, there are two main problems. First of all, we need to decide which features will be used in order to modify the output speech flow. On the other hand, it is necessary to decide which kinds of modification suit for our task. The main task of our project is to emphasize the captured behavior of the user. Therefore, accelerated fast speech and slow down slow speech would be a good approach. The problem with this approach is that we need to have an absolute measure of what *fast* and *slow* mean. Due to this, it has been decided to use the mean of an utterance to decide what has to be slowed and which parts have to be speeded up.

After that then we needed a criteria to decide the amount of acceleration or deceleration. The mean speech rate value had to stay constant, and only variations from the mean are modified. Then a function that modifies the duration of speech is used. This function translates the speech rate calculated from the input into a function that indicates the stretch that will be applied to the sound.

This function is calculated by applying a transformation function to the speech flow contour. If $sr(t)$ is the speech rate estimated, then the time transformation function $d(t)$ is:

$$d(t) = f(sr(t)) \quad (18)$$

where $f(t)$ can be expressed as:

$$f(t) = \frac{1}{(1 + e^{a(x-m)})} + 0.5 \quad (19)$$

where a is a parameter that is to be configured and m is the mean of the speaking rate contour. This function saturates at 0.5 and 1.5, therefore speech rate is only going to be reduced or increased in 50%. In Figure X there is an example of $f(t)$ for a value of $a=0.4$ and $m=2$.

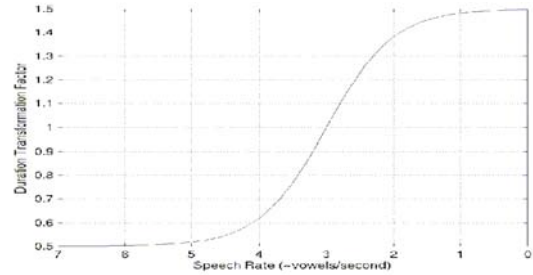


Figure 14:

In the horizontal axes the speech rate approximation is represented and on vertical axes the corresponding value for the duration modification.

APPENDIX

Implementation of prosody processing software

The main parts of the speech processing has been implemented using the PRAAT program which is distributed under GNU/GLP license. PRAAT could do the feature extraction job and the prosody modification. The prosody amplification and the time transformations have been implemented in C++ and inserted in the flow of the system by modifying Pitch and Duration files generated by PRAAT.

ACKNOWLEDGMENT

Olivier Martin is funded through a FIRST Europe fellowship from the Walloon Region (Belgium).

This work has been partially sponsored by the Spanish Government under grant TIC2002-04447-C02 (ALIADO project, <http://gps-tsc.upc.es/veu/aliado>) and the European Government under the SIMILAR Network of Excellence.

REFERENCES

- [1] Intel, "Open CV : Open source Computer Vision Library", <http://www.intel.com/research/mrl/research/opencv/>.
- [2] K.Sobottka and I.Pitas, "Segmentation and Tracking of Faces in Color Images", in Proc. of 2nd Int. Conf. on Automatic Face and Gesture Recognition 1996, pp. 236-241, Killington, Vermont, USA, 14-16 October 1996
- [3] J. Y. Bouguet, Pyramidal implementation of the Lucas-Kanade feature tracker, Intel Corporation, Microprocessor Research Labs, 1999
- [4] T. Kanade, J. Cohn, and Y. Tian, Comprehensive Database for Facial Expression Analysis, Proceedings of IEEE International Conference on Face and Gesture Recognition, pages 46-53, March, 2000
- [5] S. Krinidis, I. Pitas, "Statistical Analysis of Facial Expressions for Facial Expression Synthesis", submitted to IEEE Transactions on Pattern Analysis and Machine Intelligence, 2004
- [6] I. Kotsia, and I. Pitas, "Real time facial expression recognition from video sequences using Support Vector Machines", in Proc. of Visual Communications and Image Processing (VCIP 2005), Beijing, China, 12-15 July, 2005
- [7] J.Weston, C.Watkins, Multi-class Support Vector Machines, Technical Report CSD-TR-98-04, May 1998
- [8] Paul Boersman and David Weenink. "Praat: doing phonetics by computers", www.praat.org, July 2005.

Biologically-driven Musical Instrument

Burak Arslan, Andrew Brouse, Julien Castet, Jean-Julien Filatriau, Rémy Lehembre, Quentin Noirhomme, and Cédric Simon

Abstract—This project proposes to use the analysis of physiological signals (electroencephalogram (EEG), electromyogram (EMG), heart beats) to control sound synthesis algorithms in order to build a biologically driven musical instrument. This project took place during the eNTERFACE'05 summer workshop in Mons, Belgium. Over four weeks, specialists from the fields of brain computer interfaces and sound synthesis worked together to produce playable biologically controlled musical instruments. Indeed, a "bio- orchestra", with two new digital musical instruments controlled by physiological signals of two bio-musicians on stage, was offered to a live audience.

Index Terms—eNTERFACE'05; Electroencephalogram; EEG; Electromyogram; EMG; Biological signal; Brain Computer Interface; BCI; Music; Sound Synthesis; Sound Mapping.

I. INTRODUCTION

RECENTLY there has been much theoretical discourse about the symbiotic relationship between Art and Science. This is likely due to the fact that, for many years, Art and Science were artificially segregated as two distinct and mutually exclusive activities. Science was seen as a rigorous, methodical practice and Art as an expression of inner states, thoughts and emotions. Much recent work - including this project - attempts to develop a hybrid approach to solving complex scientific and aesthetic problems.

Advances in computer science and specifically in Human-Computer Interaction (HCI) have enabled musicians to use sensor-based computer instruments to perform music [1]. Musicians can now use data from different sensors (that reflect cardiac, or muscle activity or limb position etc.) to control sound [2], [3]. Simultaneously, advances in Brain-Computer Interface (BCI) research have shown that cerebral patterns can be used as a source of control [5]–[8]. Indeed, cerebral and conventional sensors can be used together, [4], [9], [10] with the object of producing a 'body-music' controlled according to the musician's imagination and proprioception. Some research has already been done toward integrating BCI and sound synthesis with two very different approaches. The first approach

This report, as well as the source code for the software developed during the project, is available online from the eNTERFACE'05 web site: www.enterface.net.

This research was partly funded by SIMILAR, the European Network of Excellence on Multimodal Interfaces, during the eNTERFACE05 Workshop in Mons, Belgium.

Q. Noirhomme was supported by a grant from the Région Wallonne.

Burak Arslan is with the TCTS Lab of the Faculté Polytechnique de Mons, Mons, Belgium.

Andrew Brouse is with Computer Music Research, University of Plymouth, Drake Circus, Plymouth, U.K.

Julien Castet is with Polytechnics National Institut of Grenoble, Grenoble, France.

Jean-Julien Filatriau, Rémy Lehembre, Quentin Noirhomme and Cédric Simon are with the Communications and Remote Sensing Laboratory, Université catholique de Louvain, Louvain-la-Neuve, Belgium.

is the sonification of the data [11] [12] [13]. This process can be viewed as a translation of physiological signals into sound. The second approach aims to build a musical instrument [10]. In this case, the musician tries to use his physiological signals to control intentionally the sound production. This is easy for electromyogram (EMG) or electro- oculogram (EOG) but difficult for heart sound or electroencephalogram (EEG). After long discussions at the beginning of the workshop, we did choose the second approach essentially and decided to use the sonification for additional signals to enrich the acoustical content.

In the following, we first present a short history of biological instruments and then present the architecture we developed to acquire, process and play music based on biological signals. Next we go into more detail on signal acquisition part followed by an in-depth discussion of appropriate signal processing techniques. Details of the sound synthesis implementation are then discussed along with the instruments we built. Finally, we conclude and present some future directions.

II. HISTORY

If we accept that the perception of an act as art is what makes it art, then the first instance of the use of brainwaves to generate music did not occur until 1965. Historically, brainwaves were first measured in 1924 by Hans Berger [14]. His results were verified by Matthews et al in 1934 who also attempted to sonify the measured brainwave signals in order to listen to them. This was the first example of the sonification of human brainwaves for auditory display.

In 1964, Alvin Lucier [15] had begun working with physicist Edmond Dewan, performing experiments that used brainwaves to create sound. The next year, he was inspired to compose a piece of music using brainwaves as the sole generative source. Music for Solo Performer was presented, with encouragement from John Cage, at the Rose Art Museum of Brandeis University in 1965. Lucier performed this piece several more times over the next few years, but did not continue to use EEG in his own compositions.

In the late 1960s, Richard Teitelbaum was a member of the innovative Rome-based live electronic music group Musica Elettronica Viva (MEV). In performances of Spacecraft (1967) he used various biological signals including brain (EEG) and cardiac (ECG) signals as control sources for electronic synthesizers. Over the next few years, Teitelbaum continued to use EEG and other biological signals in his compositions and experiments as triggers for nascent Moog electronic synthesizers.

Then in the late 1960s, another composer, David Rosenboom, began to use EEG signals to generate music. In 1970-

71 Rosenboom composed and performed *Ecology of the Skin*, in which ten live EEG performer-participants interactively generated immersive sonic/visual environments using custom-made electronic circuits. Around the same time, Rosenboom founded the Laboratory of Experimental Aesthetics at York University in Toronto, which encouraged pioneering collaborations between scientists and artists. For the better part of the 1970s, the laboratory undertook experimentation and research into the artistic possibilities of brainwaves and other biological signals in cybernetic biofeedback artistic systems. Many artists and musicians visited and worked at the facility during this time including John Cage, David Behrman, LaMonte Young, and Marian Zazeela. Some of the results of the work at this lab were published in the book “Biofeedback and the Arts” [16]. A more recent 1990 monograph by Rosenboom, “Extended Musical Interface with the Human Nervous System” [17], remains the definitive theoretical document in this area.

Simultaneously, Manfred Eaton was also building electronic circuits to experiment with biological signals at Orcus Research in Kansas City. He initially published an article titled “Biopotentials as Control Data for Spontaneous Music” in 1968. Then, in 1971, Eaton first published his manifesto “Bio-Music: Biological Feedback Experiential Music Systems” [18], arguing for completely new biologically generated forms of music and experience.

In France, scientist Roger Lafosse was doing research into brainwave systems and proposed, along with musique concrete pioneer Pierre Henry, a sophisticated live performance system known as CorticalArt (art from the cerebral cortex). In a series of free performances done in 1971, along with generated electronic sounds, one saw a television image of Henry in dark sunglasses with electrodes hanging from his head, projected so that the content of his brainwaves changed the color of the image according to his brainwave patterns.

At the same time, in the early 1970s, Jacques Vidal, a computer science researcher at UCLA, began working to develop the first direct brain-computer interface (BCI) [20]. In 1990 two scientists, Benjamin Knapp and Hugh Lusted [19], began working on a computer interface called the BioMuse. It permitted a human to control certain computer functions via bioelectric signals primarily via EMG. In 1992, Atau Tanaka [1] was commissioned by Knapp and Lusted to compose and perform music using the BioMuse as a controller. Tanaka continued to use the BioMuse, primarily as an EMG controller, in live performances throughout the 1990s.

III. ARCHITECTURE

We intend to build a robust architectural framework that could be reused with other biological data, other methods of analysis and other types of instruments. Therefore the signal acquisition, the signal processing and the sound synthesis are operated on different virtual machines that communicate by the network (Fig. 1). The data from the different modalities are recorded on different machines. Once acquired, the data are sent to a Simulink [21] program. Then they are processed before being sent to the musical instruments, and the sound spatialization and visualization routines via Open Sound

Control. The musical instruments are build with Max/MSP. Below is an outline of the main software and data exchange architecture.

A. Software

1) *Matlab and Simulink*: Biosignal analysis is achieved with various methods including wavelet analysis and spatial filtering. Due to the flexibility of Matlab [21] programming, all the algorithms are written in Matlab code. However, since the signal acquisition from the EEG cap is made in C++ we first used a method in C++ that called the Matlab codes. EEG activity varies from a person to another. Thus in order to have a good adaptation to all subjects and change parameters like frequency bands online, we implemented our sources in a Simulink block diagram using Level-2 M file S-functions with tuneable parameters. This allows us to adapt online to the incoming signals from the subjects scalp. Subsequently, we can proceed with a real-time, manually controlled, adaptive analysis. Simulink offers many possibilities in terms of visualization. For example, we used the virtual reality toolbox in order to have some feedback and help the user control his/her EEG. The graphical interface used here is quite simple and consists of a ball moving to the right or to the left whether the user is moving his right or left hand.

2) *Max/MSP*: Max/MSP [23] is a software programming environment optimized for flexible real-time control of music systems. It was first developed at IRCAM by Miller Puckette as a simplified front-end controller for the 4X series of mainframe music synthesis systems. It was further developed as a commercial product by David Zicarelli [28] and others at Opcode Systems and Cycling 74 [29]. It is currently the most popular environment for programming of real-time interactive music performance systems.

Max/MSP is interesting to use in that it is a very mature, widely accepted and supported environment. The result of this is that few problems are encountered which cannot be resolved simply with recourse to the many available support resources. There are, however, some concerns about its continued use in an academic environment where open-source software systems are increasingly preferred or even required. There are other open-source environments which could be more interesting in the long-term especially in an academic context: Pure Data and jMax are both open-source work-alike software implementations which although not as mature as Max/MSP are nonetheless very usable. SuperCollider is another, text-based, programming environment which is also very powerful but is somewhat more arcane and difficult to program.

B. Data Exchange

Data are transferred from one machine to another with the UDP protocol. We chose it mainly for its better real-time capability. To communicate with the musical instrument we use a specific protocol one level higher than UDP: open sound control (OSC) [22].

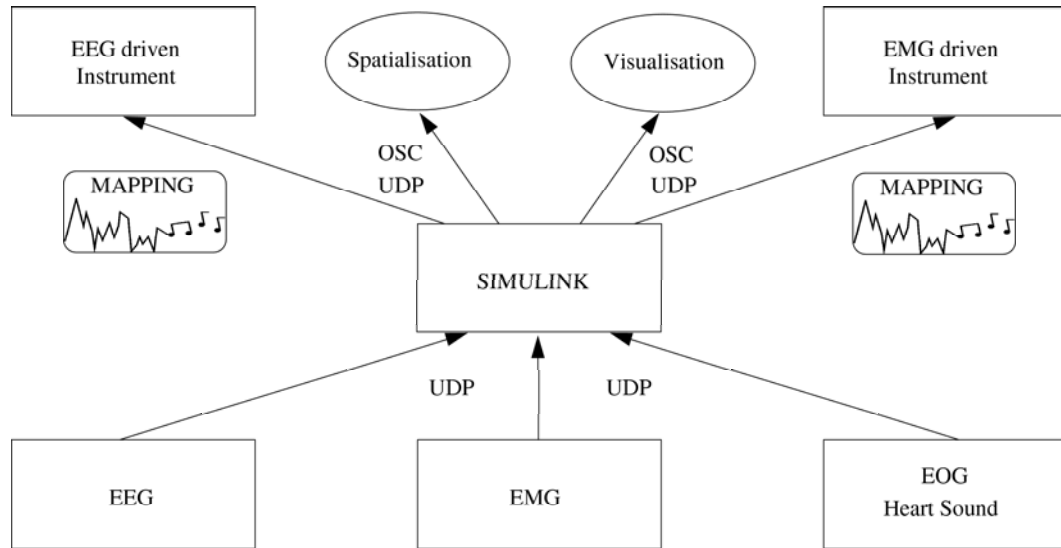


Fig. 1. System architecture

Open Sound Control: OSC was conceived as a protocol for the real-time control of computer music synthesizers over modern heterogeneous networks. Its development was informed by shortcomings experienced with the established MIDI standard and the difficulties in developing a more flexible protocol for effective real-time control of expressive music synthesis. Various attempts had been made to produce a replacement for the MIDI protocol such as ZIPI which was proposed and then abandoned. OSC was first proposed by Matthew Wright and Adrian Freed in 1997. Since that time its use and development have grown such that it is becoming very widely implemented in software and hardware designs (although, still not as widespread as MIDI). Although it can function in principle over any appropriate transport layer such as WiFi, serial, USB or other data network, current implementations of OSC are optimized for UDP/IP transport over Fast Ethernet in a Local Area Network. For our project, we used OSC to transfer data from Matlab (running on a PC with either Linux or Windows OS) towards Max/MSP (running on a Macintosh OSX).

IV. DATA ACQUISITION

Four types of signals are recorded with associated captors: EEG, EMG, heart sounds and EOG (see Fig. 2). EMG, EOG and heart sounds are acquired on one machine and EEG on another.

A. Electroencephalogram (EEG)

EEG data (Fig. 3) are recorded at 64 Hz on 19 channels with a DTI cap [24]. Data are filtered between 0.5 and 30 Hz. Channels are positioned following the 10-20 international system and Cz is used as reference. The subject sits in a comfortable chair and is asked to concentrate on different tasks. The recording is done in a normal office environment, e.g. a noisy room with people working, speaking and with music. The environment is not free from electrical noise as

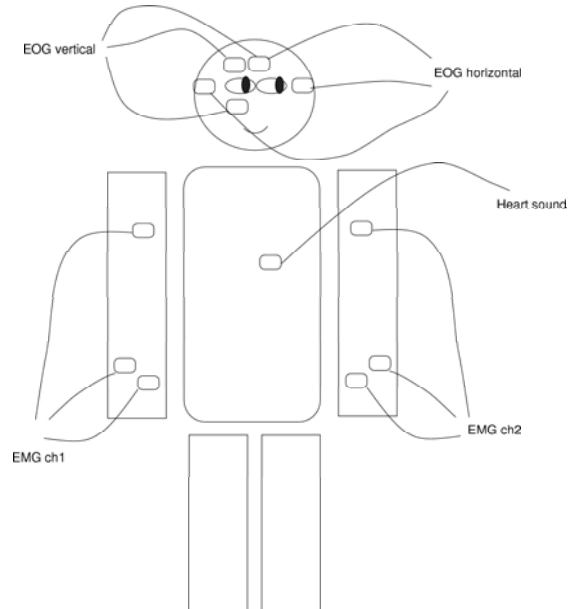


Fig. 2. Application of multiple electrodes and transducers

there are many computers, speakers, screen, microphones and lights around.

B. Electromyogram (EMG), Heart sounds and Electro-oculogram (EOG)

To record the EMG (Fig. 4) and heart sounds (Fig. 5), three amplifiers of Biopac MP100 system [25] were used. The amplification factor for the EMG was 5000 and the signals were filtered between 0.05-35 Hz. The microphone channel had 200 gain and DC-300 Hz bandwidth. Another 2 channel amplifier, ModularEEG [26] is used to collect the EOG signals (Fig. 6). This amplifier has 4000 gain and 0.4-60 Hz passband.

For real time capabilities, these amplified signals are fed to the National Instruments DAQPad 6052e [27] analog-

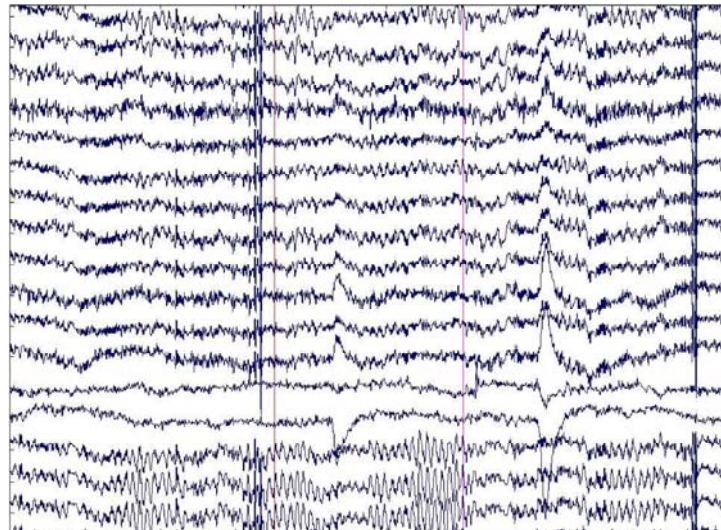


Fig. 3. EEG signals

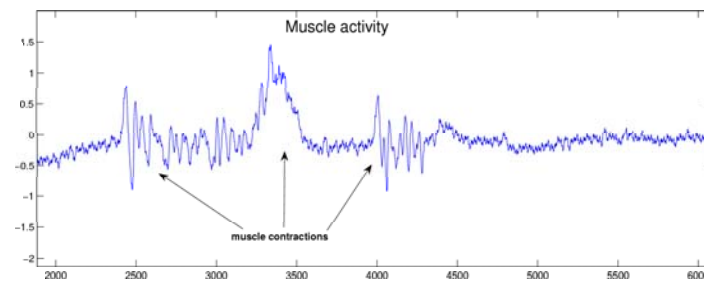


Fig. 4. EMG signal

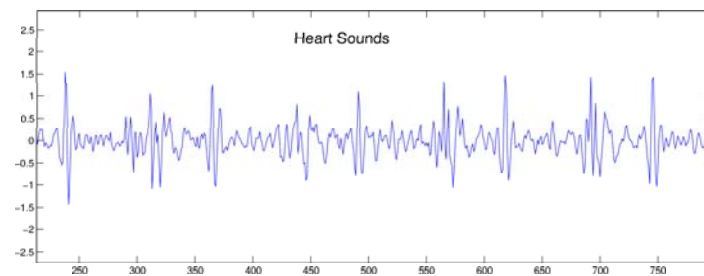


Fig. 5. Heart sound signal

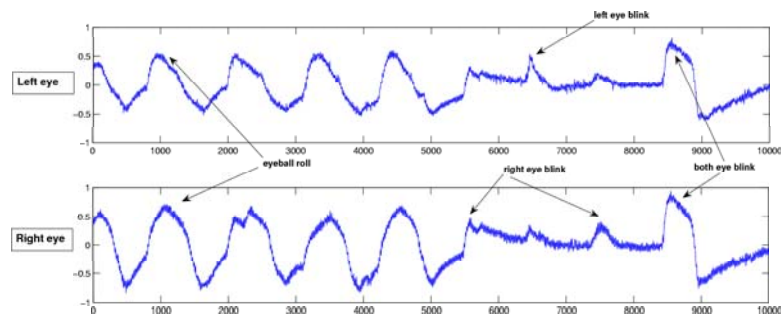


Fig. 6. EOG signal

digital converter card that uses the IEEE 1394 port. Thus, the data can be acquired, processed and transferred to the musical instruments using Matlab environment and the Data Acquisition toolbox.

Disposable ECG electrodes were used for both EOG and EMG recordings. The sounds were captured using the Biopac BSL contact microphone.

V. BIOSIGNAL PROCESSING

The aim of this work is to control sound and synthesize music using parameters derived from measured biological signals such as: EEG, EOG, EMG and heart sounds. We therefore have tested different techniques to extract parameters giving meaningful control data to drive musical instruments. We mainly concentrated on EEG signal processing as it is the richest and most complex bio-signal. The musician normally has better conscious control over bio-signals other than EEG and therefore only basic signal processing is done in these cases. The data acquisition program samples blocks of EMG or EOG data of 100 ms duration, and then analyzes this data. It calculates the energy for the EOG and EMG channels, and sends this information to the related instruments. The heart sound itself is directly sent to the instruments to provide a background motif, which can be also used to control the rhythmic structure. The waveform can also be monitored on the screen in real-time.

Two kinds of EEG analysis are done. The first one focuses on the detection of a users intent. It is based on the work being done in the BCI community [7]. A second approach looks at the origin of the signal and at the activation of different brain areas. The musician has less control over results in this case. At the end of this section there are more details on both of these EEG analysis approaches (Fig. 7).

A. Detection of User's Intent

To detect different brain states we used the spatialisation of the activity and the different rhythms present in this activity. Indeed, each part of the brain has a different function and each human being may present specific rhythms at different frequencies. For example, three main rhythms are of great interest:

- 1) Alpha rhythm: usually between 8-12 Hz, this rhythm describes the state of awareness. If we calculate the energy of the signal using the occipital electrodes, we can evaluate the awareness state of the musician. When he closes his eyes and relaxes the signal increases. When the eyes are open the signal is low.
- 2) Mu rhythm: This rhythm is also reported to range from 8 to 12 Hz but this band can vary from one person to another, sometimes between 12-16 Hz. The mu rhythm corresponds to motor tasks like moving the hands or legs, arms, etc. We use this rhythm to distinguish left hand movements from right hand movements.
- 3) Beta rhythm: Comprised of energy between 18-26 Hz, the characteristics of this rhythm are yet to be fully understood but it is believed that it is also linked to motor tasks and higher cognitive function.

Therefore the well-known wavelet transform [30] is a technique of time-frequency analysis perfectly suited for the task detection. Each task can be detected by looking at specific bandwidth on specific electrodes.

This operation, implemented with sub-band filters, provides us with a filter bank tuned to the frequency ranges of interest. We tested our algorithm on two subjects with different kinds of wavelets: Meyer wavelet, 9-7 filters, bi-orthogonal spline wavelet, symlet 8 and Daubechey 6 wavelets. We finally chose the symlet 8 which gave better overall results. Once the desired rhythms are obtained, different forms of analysis are possible.

At the beginning we focused on eye blink detection and α band power detection because both are easily controllable by the musician. We then wanted to try more complex tasks such as those used in the BCI community. These are movements and imaginations of movements, such as hand, foot or tongue movements, 3D spatial imagination or mathematical calculation. The main problem is that each BCI user needs a lot of training to improve his control of the task signal. Therefore we decided to use only right and left hand movements first and not the more complex tasks which would have been harder to detect. Since more tasks also means more difficult detection, they are the only tasks used in this project. Two different techniques were used: Asymmetry ratio and spatial decomposition.

1) *Eye blinking and α band*: Eye blinking is detected on Fp1 and Fp2 electrodes in the 1-8Hz frequency range by looking at increase of the band power. We process the signals from electrodes O1 and O2 -occipital electrodes- to extract the power of the alpha band.

2) *Asymmetry ratio*: Consider we want to distinguish left from right hand movements. It is known that motor tasks activate the motor cortex area. Since the brain is divided into two hemispheres that control the two sides of the body it is possible to recognize when a person moves on the left or right side. Let C3 and C4 be the two electrodes positioned on the cortex, the asymmetry ratio can be written as:

$$\Gamma_{FB} = \frac{P_{C3,FB} - P_{C4,FB}}{P_{C3,FB} + P_{C4,FB}} \quad (1)$$

where $P_{Cx,FB}$ is the power in a specified frequency band (FB), i.e. the mu frequency band. This ratio has values between 1 and -1. Thus it is positive when the power in the left hemisphere (right hand movements) is higher than the one in the right hemisphere (left hand movements) and vice-versa.

The asymmetry ratio gives good results but is not very flexible and cannot be used to distinguish more than two tasks. This is why it is necessary to search for more sophisticated methods which can process more than just two electrodes as the asymmetry ratio does.

3) *Spatial decomposition*: Two spatial methods have proven to be accurate: The Common Spatial Patterns (CSP) and the Common Spatial Subspace Decomposition (CSSD) [31], [32]. We will shortly describe here the second one (CSSD): This method is based on the decomposition of the covariance matrix grouping two or more different tasks. Only the simple case of two tasks will be discussed here. It

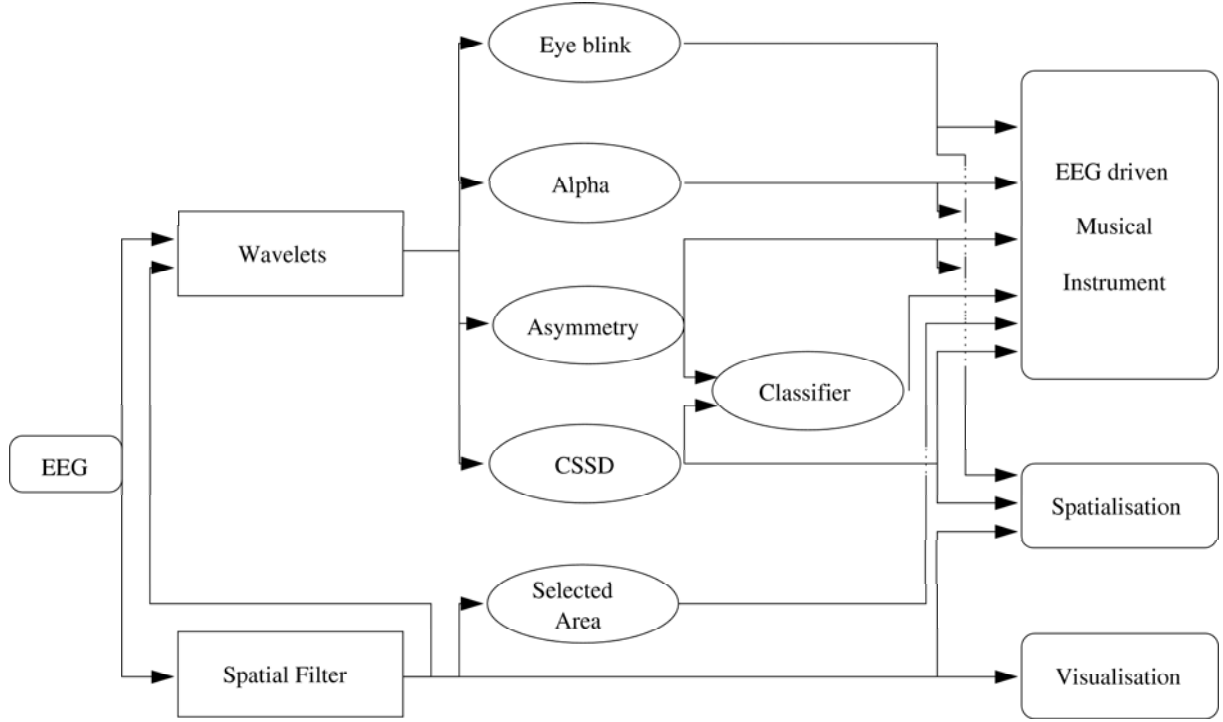


Fig. 7. EEG processing, from recording (left) to play (right).

is important to highlight the fact that this method needs a learning phase where the user executes the two tasks.

The first step is to compute the autocovariance matrix for each tasks. Lets take one signal X of dimension $N \times T$ for N electrodes and T samples. Decomposing X in X_A et X_B , A and B being two different tasks, we can obtain the autocovariance matrix for each task:

$$R_A = X_A X_A^T \quad \text{and} \quad R_B = X_B X_B^T \quad (2)$$

We now extract the eigenvectors and eigenvalues from the R matrix that is the sum of R_A and R_B :

$$R = R_A + R_B = U_0 \lambda U_0^T \quad (3)$$

We can now calculate the spatial factors matrix W and the whitening matrix P :

$$P = \lambda^{-1/2} U_0^T \quad \text{and} \quad W = U_0 \lambda^{1/2} \quad (4)$$

If $S_A = P R_A P^T$ and $S_B = P R_B P^T$, these matrices can be factorized:

$$S_A = U_A \Sigma_A U_A^T \quad S_B = U_B \Sigma_B U_B^T \quad (5)$$

Matrices U_A et U_B are equal and the sum of their eigenvalues is equal to 1, $\Sigma_A + \Sigma_B = I$. Σ_A et Σ_B can be written thus:

$$\Sigma_A = \text{diag}[\underbrace{1 \dots 1}_{m_a}, \underbrace{\sigma_1 \dots \sigma_{m_c}}_{m_c}, \underbrace{0 \dots 0}_{m_b}] \quad (6)$$

$$\Sigma_B = \text{diag}[\underbrace{0 \dots 0}_{m_a}, \underbrace{\delta_1 \dots \delta_{m_c}}_{m_c}, \underbrace{1 \dots 1}_{m_b}] \quad (7)$$

Taking the first m_a eigenvector from U , we obtain U_a and we can now compute the spatial filters and the spatial factors:

$$S P_a = W U_a \quad (8)$$

$$S F_a = U_a^T P \quad (9)$$

We proceed identically for the second task, but taking this time the last m_b eigenvectors. Specific signal components of each task can then be extracted easily by multiplying the signal with the corresponding spatial filters and factors. For the task A it gives:

$$\hat{X}_a = S P_a S F_a X \quad (10)$$

A support vector machine (SVM) with a radial basis function was used as a classifier.

4) *Results:* The detection of eye blinking during off-line and realtime analysis was higher than 95%, with a 0.5s time window. For hand movement classification with spatial decomposition, we chose to use a 2s time window. A smaller window significantly decreases the classification accuracy. The CSSD algorithm needs more training data to achieve a good classification rate so we decided to use 200 samples of both right hand and left hand movements, each sample being a 2s time window. Thus, we used an off-line session to train the algorithm. However each time we used the EEG cap for a new session, the electrode locations on the subject's head changed. Performing a training session one time and a test session another time gave poor results so we decided to develop new code in order to do both training and testing in one session. This had to be done quite quickly to ensure the user's comfort.

We achieved an average of 90% good classifications during off-line analysis, and 75% good classifications during real-time recording. Real-time recording accuracy was a bit less than expected. (This was probably due to a less-than-ideal environment - with electrical and other noise - which is not

conductive to accurate EEG signal capture and analysis.) The asymmetry ratio gave somewhat poorer results.

B. Spatial Filters

EEG is a measure of electrical activities of the brain as measured on the external skull area. Different brain processes can activate different areas. Thus, knowing which areas are active can inform about active cerebral processes. Discovering these active areas is rather a difficult task, as many source configurations can lead to the same EEG recording. Noise in the data further complicates the problem. The ill-posedness of the problem leads to many different methods based on different hypotheses to get a unique solution. In the following, we present the methods - based on forward and inverse problems - and the hypothesis we propose to solve the problem in real time.

1) *Forward Problem, head model and solution space:* If X is a $N \times 1$ vector containing the recorded potential with N representing the number of electrodes. S is an $M \times 1$ vector of the true source current with M the unknown number of sources. G is the leadfield matrix which links the source location and orientation to the electrodes location. G depends of the head model. n is the noise. We can write

$$X = G S + n \quad (11)$$

X and S can be extended to more than one dimension to take time into account. S can either represent few dipoles (dipole model) with $M \leq N$ or represent the full head (image model - one dipole per voxel) with $M \gg N$. In the following we will use the latter model.

The forward problem is to try and find the potentials X on the scalp surface knowing the active brain sources S . This approach is far simpler than the inverse approach and its solution is the basis of all Inverse problem solutions.

The leadfield G is based on the Maxwell equations. A finite element model based on the true subject head can be used as lead field but we prefer to use a 4-spheres approximation of the head. It is not subject dependent and less computationally expensive. A simple method consists of seeing the multi-shell model as a composition of single-shells -much as Fourier uses functions as sums of sinusoid [33]. The potential v measured at electrode position r from a dipole q in position r_q is

$$v(r, \mu_1 r_q, \lambda_1 q) + v^1(r, \mu_2 r_q, \lambda_2 q) + v^1(r, \mu_3 r_q, \lambda_3 q) \quad (12)$$

λ_i and μ_i are called Berg's parameters [33]. They have been empirically computed to approximate three and four-shell head model solution.

When we are looking for the location and orientation of the source, a better approach consists of separating the non-linear search for the location and the linear one for the orientation. The EEG scalar potential can then be seen as a product $v(r) = k^t(r, r_q)q$ with $k(r, r_q)$ a 3×1 vector. Therefore each single shell potential can be computed as [34]

$$v^1(r) = ((c_1 - c_2(r, r_q))r_q + c_2\|r_q\|^2 r) \cdot q$$

with

$$c_1 \equiv \frac{1}{4\pi\sigma\|r_q\|^2} \left(2 \frac{d \cdot r_q}{\|d\|^3} + \frac{1}{\|d\|} - \frac{1}{\|r\|} \right) \quad (13)$$

$$c_2 \equiv \frac{1}{4\pi\sigma\|r_q\|^2} \left(\frac{2}{\|d\|^3} + \frac{\|d\| + \|r\|}{\|r\|F(r, r_q)} \right) \quad (14)$$

$$F(r, r_q) = \|d\|(\|r\|\|d\| + \|r\|^2 - (r_q \cdot r)) \quad (15)$$

The brain source space is limited to 361 dipoles located on an half-sphere just below the cortex in a perpendicular orientation to the cortex. This is done because the activity that we are looking is concentrated on the cortex; the activity recorded by the EEG is mainly cortical activity and the limitation of the source space considerably reduces the computation time.

2) *Inverse Problem:* The inverse problem can be formulated as a Bayesian inference problem [35]

$$p(S|X) = \frac{p(X|S)p(S)}{p(X)} \quad (16)$$

where $p(x)$ stands for probability distribution of x . We thus look for the sources with the maximum probability. Since $p(X)$ is independent of S it can be considered as a normalizing constant and can be omitted. $p(S)$ is the prior probability distribution of S and represents the prior knowledge we have about the data. This is modified by the data through the posterior probability distribution $p(X|S)$. This probability is linked to the noise. If the noise is gaussian - as everybody assumed - with zero mean and covariance matrix C_n

$$\ln p(X|S) = (X - GS)^t C_n^{-1} (X - GS) \quad (17)$$

where t stands for transpose. If the noise is white, we can rewrite equation (17) as

$$\ln p(X|S) = \|X - GS\|^2 \quad (18)$$

In case of zero mean gaussian prior $p(S)$ with variance C_S , the problem becomes

$$\begin{aligned} & \argmax(\ln p(S|X)) \\ &= \argmax(\ln p(X|S) + \ln p(S)) \\ &= \argmax((X - GS)^t C_n^{-1} (X - GS) + \lambda S^t C_S S) \end{aligned}$$

where the parameter λ gives the influence of the prior information. And the solution is

$$\hat{S} = G^t C_n^{-1} (G^t C_n^{-1} G + \lambda C_S^{-1})^{-1} X \quad (19)$$

For a full review of methods to solve the Inverse Problem see [35]–[37].

Methods based on different priors were tested. Priors ranged from the simplest -no prior information- to classical prior such as the laplacian and to a specific covariance matrix. The well-know LORETA approach [37] showed the best results on our test set. The LORETA [37] looks for a maximally smooth solution. Therefore a laplacian is used as a prior. In (19) C_s is a laplacian on the solution space and C_n is the identity matrix.

To enable real time computation, leadfield and prior matrices in (19) are pre-computed. Then we only multiply the pre-computed matrix with the acquired signal. Computation time is less than 0.01s on a typical personal computer.

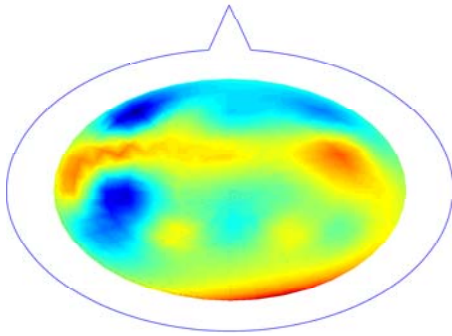


Fig. 8. Derived current at the surface of the brain. The scale is going from blue the more negative potential to red the more positive potential

3) *Results and Application:* In the present case of a BCMI, the result can be used for three potential applications: the visualization process, a pre-filtering step and a processing step.

The current of the 361 dipoles derived using the inverse method is directly used in the visualization process. The current on every point of the half-sphere is interpolated from the dipole currents. The result is projected on a screen (see Fig. 8).

The result of the inverse solution could be used as a pre-filtering step in the classification process. Instead of using the 18 electrode signals, the 361 dipole signals can be used. We did not have enough time to test this approach.

The results of the inverse solution reflect the brain activity. Therefore it could be used as direct control data for our musical instrument. Four brain areas were selected. They were the frontal area, the occipital area and both left and right sensori-motor and motor areas. The frontal area is generally linked to cognition and memory processes. Left and right sensori-motor and motor cortex areas are linked to movement and imagination of movement in the right and left parts of the body respectively. The occipital area is involved in visualization processes. For every area, we compute the mean of the source signal in the area. The mean of each area is then scaled and sent as control data for the musical instruments. The dipoles inside each area were selected on a visual basis in order to adequately cover the relevant areas (Fig. 9).

VI. SOUND SYNTHESIS

A. Introduction

1) *Sound Synthesis:* Artificial synthesis of sound is the creation, using electronic and/ or computational means, of complex waveforms, which, when passed through a sound reproduction system can either mimic a real musical instrument or represent the virtual projection of an imagined musical instrument. This technique was first developed using digital computers in the late 1950s and early 1960s by Max Matthews at Bell Labs. It does have antecedents, however, in the musique concrete experiments of Pierre Schaeffer and Pierre Henry and in the TelHarmonium of Thaddeus Cahill

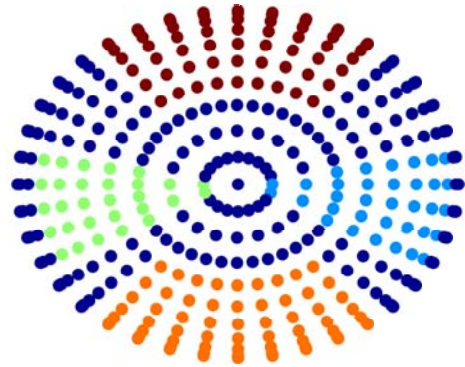


Fig. 9. Dipoles are set in 4 areas. Dark blue dipoles are outside all the area. Light blue dipoles are in right sensorimotor and motor cortex area. Green dipoles are in left sensorimotor and motor cortex area. Orange dipoles are in occipital area. Brown dipoles are in frontal area

amongst others. The theory and techniques of sound synthesis are now widely developed and are treated in depth in many well-known sources.

The chosen software environment, Max/MSP, makes available a wide palette of sound synthesis techniques including: additive, subtractive, frequency modulation, granular etc. With the addition of 3rd party code libraries (externals) Max/MSP can also be used for more sophisticated techniques such as physical modelling synthesis.

2) *Mapping:* The very commonly used term mapping refers, in the instance of virtual musical instruments, to mathematical transformations which are applied to real-time data received from controllers or sensors so that they may be used as effective control for sound synthesis parameters. This mapping can consist of a number of different mathematical and statistical techniques. To effectively implement a mapping strategy one must understand well both the ranges and behavior of the controller or sensor data and the synthesis parameters which are to be controlled. For our purposes, it is most important to be mindful of the appropriate technique to be used in order to achieve the desired results.

A useful way of thinking about mapping is to consider its origin in the art of making cartographic maps of the natural world. Mapping thus is forming a flat, virtual representation of a curved, spherical real world which enables that real world to be effectively navigated. Implicit in this is the process of transformation or projection which is necessary to form the virtual representation. This projection is not a transparent process but can involve decisions and value judgements. The commonly-used Mercator projection of the world, for example, gives greater apparent land mass and thus import to the western and northern parts of the world where that projection was initially developed and used. Buckminster Fuller attempted to redress this issue with his Geodesic projection of the world which was felt to be a more accurate representation of the earth's surface.

Thus, to effectively perform a musically satisfying mapping, we must understand well the nature of our data sources (sen-

sors and controllers), the nature of the sounds and music we want to produce (including intrinsic properties and techniques of sound synthesis, sampling, filtering and DSP)

This poses significant problems in the case of biologically controlled instruments in that it is not possible to have an unambiguous interpretation of the meanings of biological signals whether direct or derived. There is some current research in cognitive neuroscience which may indicate directions for understanding and interpreting the musical significance of encephalographic signals at least.

A simple example is the alpha rhythm or more correctly alpha spectrum of the EEG. It is well known that strong energy in the frequency band (8-13 Hz) indicates a state of unfocused relaxation without visual attention in the subject. This has commonly been used as a primary controller in EEG-based musical instruments such as Alvin Luciers "Music for Solo Performer", where strong EEG will directly translate to increased sound intensity and temporal density. If this is not the desired effect then consideration has to be given to how to transform the given data into the desired sound or music.

At the end of the workshop, a musical bio-orchestra, composed by two new digital musical instruments controlled by two bio-musicians on stage (Fig. 10), offered a live performance to a large audience. The first instrument was a midi instrument based on additive synthesis and controlled by musician's electroencephalograms plus an infrared sensor. The second one, driven by electromyograms of a second bio-musician, processed accordion samples recorded in live situation by granulation and filtering effects. Furthermore biological signals managed the spatialized diffusion over eight loudspeakers of sound produced by both previous instruments and the visual feedback. This was controlled by EEGs of the first bio-musician. We here present details of each of these instruments.

B. Instrument 1 : a new interface between brain and sound

EEG analysis can detect many things about eyes and movements, but it needs training to give good results. For this interface, we used the following controls (Fig. 11):

- right or left body part movement (Mu bandwidth)
- eyes are open or closed (Alpha bandwidth)
- the average activity of brain (Alpha bandwidth)

The MAX/MSP patch is based upon these parameters. The sound synthesis is done with a plug-in from Absynth which is software controlled via the MIDI protocol. The patch creates MIDI events controlling the synthesis which is in particular composed of three oscillators, three Low Frequency Oscillators, and three notch filters. There are two kinds of note trigger:

- a cycle of seven notes
- a trigger of single note

This work needed high-level treatment, so pitch is not controlled continuously. The mapping between sound parameters and control parameters is explained below.

Regarding the first kind of note trigger, the cycle of notes begin when the artist opens his eyes for the first time. Then,

there is another type of control using EEG analysis, when the artist thinks about right or left body movements, he controls the direction of cycle rotation and the panning of the result. The succession of notes is subjected of two randomized variations, the note durations and the delta time between each note.

Regarding the second note trigger, alpha bandwidth is converted to a number between 0 and 3, and is divided into three parts:

- 0 to 1 : this part is divided into five sections, one note is attributed to each section and the time properties are given by the dynamics of the alpha variations
- 1 to 2 : represents the variation of the Low Frequency Oscillator (LFO) frequency
- over 2 : the sound is stopped

The EEG analysis for these controls happens over time, and to have an instantaneous control, an infrared sensor controller was added. According to the distance between his hand and the sensor, the artist can control:

- the rotation speed of the cycle, using the right hand
- the frequency of the two other LFO, using the left hand

EEG analysis can detect if the artist moves his right or his left hand, so this one sensor is the source of two kinds of control.

As you can see in the elaboration of this patch, it is already an aesthetic choice in that the performer decides the harmony before playing. This is not the only solution, but in the performance which was done, it has proved to be a good solution (Fig. 10).

Another advantage of this patch is its modularity. An artist can depend on it to create lots of different sound results. The patch is a real interface between a synthesis software using MIDI protocol and an EEG analysis with Matlab.

Results: The aim of this work was to create an instrument commanded by EEG signals, but can we actually talk about a musical instrument? Instrumental relationships are always linked with gestures. Here no physical interaction is present. Further, the complexity of the interaction with a traditional musical instrument, like a guitar, assigns an important power of manipulation to the artist. To be interesting from an artistic point of view, a musical instrument must give a large expressive space to the artist; this was a big challenge in our case, and it seems to have been partially effective. In this instrument, the relation between the artist and his production is really peculiar because it acts on two levels: the musician interacts with sound production by means of his EEGs but the produced sound also has a feedback influence on the mental state of the musician. Future work could turn towards the biofeedback influence of sound. When the musician tries to control his brain activities, the sound perturbs him. What kind of influence could there be?

C. Instrument 2 : Real-time granulation and filtering on accordion samples

In our second instrument, sound synthesis is based on the real-time granulation and filtering of accordion samples recorded in live situation by the bio-digital musician. During the demonstration, the musician started his performance by

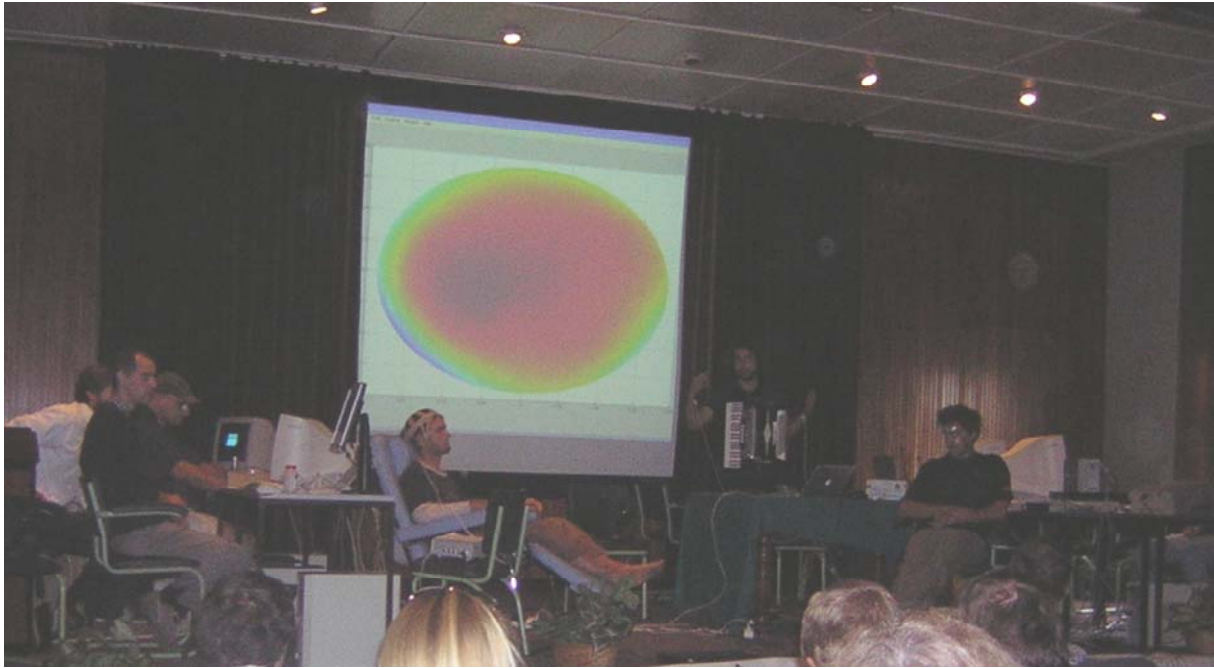


Fig. 10. Concert during eINTERFACE 2005 Workshop

playing and recording few seconds of accordion that will then be processed in real-time. Sound processing was implemented thanks to several Max-MSP objects and controlled by means of data extracted from electromyograms (EMGs) measuring both arms muscles contraction of the musician (Fig. 12). An additional MIDI surface control was also used to extend the possibilities of mapping.

1) *Granulation*: Granulation techniques [38] split an original sound into very small acoustic events called grains of 50 ms duration or less, and reproduces them in high densities ranging from several hundred to several thousand grains per second. A lot of transformations (time stretching, pitch shifting, backward reading) on the original sound are made possible with this technique and a large range of very strange timbres, far from the original, can be obtained in this way. In our instrument, the granulation was achieved by MSP object `munger~`, released as part of the free Max/MSP toolkit `PeRColate` developed by Trueman and DuBois [39]. `Munger~` takes an incoming audio signal input and granulates it, breaks it up into small grains which are layered, mixed and transposed as requested, creating cloud-like textures of varying densities. Furthermore, the `munger~` object has several arguments that enable to modify the resulting granulated sound. In order to give the musician the ability, we choose to control three of them:

- the grains size (in ms)
- the pitch shifting : this parameter control the playback speed and allows to transpose all outgoing grains by a multiplier factor.
- the pitch shifting variation (factor between 0 and 1) : `munger~` enables to vary randomly the pitch shifting factor : more precisely, the "grain pitch variation" parameter will control how far into a predefined scale the

`munger~` will look for the pitch shifting factor. Increasing this parameter has a strong effect on the resulting sound by making it very turbulent. To enhance this turbulence sensation, we coupled this parameter with swirling spatialization effect. This was the only spatialization effect controlled by the EMG musician, the rest of the spatialization being driven by EEG analysis.

In term of mapping, the performer selected the synthesis parameter he wanted to vary thanks to the midi foot controller and this parameter was then modulated according to the contraction of his arm muscles, measured by electromyograms. The contraction of left arm muscles allowed choosing either to increase or decrease the selected parameter, whereas the variation of the parameter, between predefined range, was directly linked to right arm muscle tension.

2) *Flanging*: We tried some of the most widely used filtering effects in audio processing (chorus, delay, phase shifting etc) and finally we chose to integrate flange effect as filter processing in this first version of our instrument. Flanging is created by mixing a signal with a slightly delayed copy of itself, where the length of the delay, less than 10 ms, is constantly changing (Fig. 13). Instead of creating an echo, the delay has a filtering effect on the signal, and this effect creates a series of notches in the frequency response. This varying delay in the flanger creates some pitch modulation (warbling pitch).

In order to process accordion samples by flange effect, we used in our instrument the example of flange effect provided in the MSP tutorial. The musician chose among different predefined parameters configurations. He had also the ability to modulate each parameter (depth, feedback gain, LFO frequency) separately via his arm muscles contraction, by the same way than for the granulation parameters.

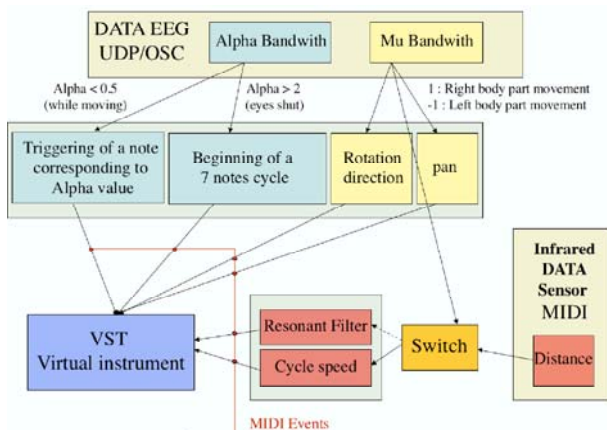


Fig. 11. the functional instrument diagram

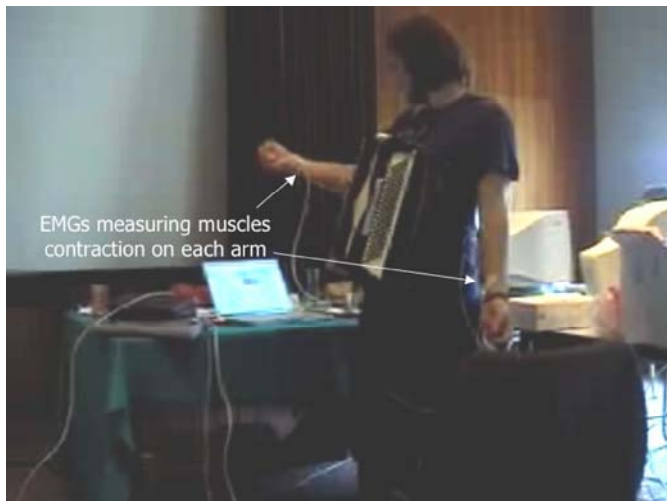


Fig. 12. Bio-musician controlling his musical instrument by means of his muscles contraction

3) *Balance dry/wet sounds*: During the performance, the musician chose to vary whether or not the sound processing parameters (granulation or flange parameters). When the musician does not act on these parameters, he had the possibility to control the intensities of dry and wet sounds with the contraction of his left and right arm respectively. This control gave the musician the ability to cross-fade original sound with the processed one by means of very expressive gestures.

4) *Results*: Very interesting sonic textures, nearer or farther from original accordion sound, have been created by this instrument. Granulation gave the sensation of clouds of sound, whereas very strange sounds, reinforced by spatialisation effects on eight loudspeakers, were obtained using certain parameter configurations of the flange effect. A pleasant way to use this instrument was to superimpose live accordion notes on these synthesized sonic soundscapes such as to create a hyper-accordion. Using arm muscle contractions, measured as EMGs, to control synthesis parameters gave worthwhile results because sound production was controlled via expressive gestures.

5) *Future*: At the end of the workshop, the design of this bio-instrument was just finished. Thus, as with a traditional musical instrument, the first thing to do will be to practice the instrument in order to properly learn it. These training sessions will especially aim to improve the mapping between sound parameters and gestures, by making it simpler and more intuitive.

Regarding the sound processing/synthesis itself, trying other kinds of sound processing could give interesting results. Among the difficulties we encountered in designing this instrument controlled by EMG, was the lack of available control parameters extracted from EMGs analysis, hence the need of an additional midi controller to build an entire instrument. Furthermore, this type of mapping relied on arm muscle contractions, which could also be achieved by means of data gloves [40] ; which is why it would be very interesting to add EMGs measuring muscles contraction in other body areas (legs, shoulders, neck) in order to give a real added richness to this bio-instrument.

D. Spatialization and Localization

The human perception of the physical location of sound sources within a given physical sound environment are due to a complex series of cues which have evolved according to the physical behavior of sound in real spaces. These cues can include: intensity, including right-left balance, relative phase, early reflections and reverberation, Doppler shift, timbral shift and many other factors which are actively studied by researchers in auditory perception.

The terms 'spatialization' and 'localization' are germane to the study and understanding of this domain. The term 'spatialization' refers to the creation of a virtual sound space using electronic techniques (analogue or digital) and sound reproduction equipment (amplifiers and speakers) to either mimic the sound-spatial characteristics of some real space or present a virtual representation of an imaginary space reproduced via electronic means. The term 'localization' refers

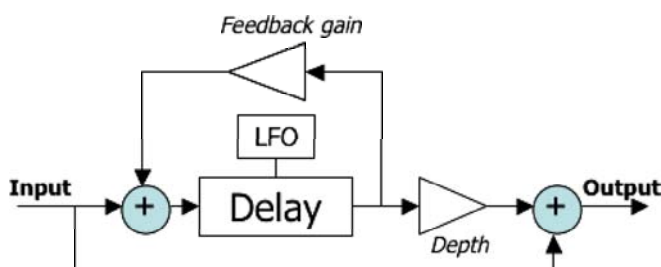


Fig. 13. Diagram of flanger effect. The delay is varying with time thanks to a low frequency oscillator (LFO) whose frequency is user-controllable. The depth parameter allows to control how much of the delayed signal is added to the original one. Feedback gain specifies the amount of feedback signal to be added to the input signal ; a large amount of feedback will create a very 'metallic' and 'intense' sound.

to the placement of a given sound source within a given spatialized virtual sound environment using the techniques of spatialization. Given the greatly increased real-time computational power available in today's personal computers, it is now possible to perform complex and subtle spatialization and localization of sounds using multiple simultaneous channels of sound reproduction (four or more). Thus spatialization is the creation of virtual sound environments and localization is the placement of given sounds within that virtual environment.

The implementation of a system for the localization of individual sound sources and overall spatialization in this project was based around an 8 channel sound reproduction system. Identical loudspeakers were placed equidistant in a circular pattern around a listening space all at the same elevation - approximately at ear level. Sounds were virtually placed within the azimuth of this 360 degree circular sound space by the use of mixing software which approximates an equal-power panning algorithm. The amplitude of each virtual sound source can be individually controlled. Artificial reverb can be added to each sound source individually in order to simulate auditory distance. Finally, each individual sound source can be placed at any azimuth and panned around the circle in any direction and at any speed.

Future implementations of this software will take into account more subtle aspects of auditory localization including timbral adjustments and Doppler effects.

E. Visualization

In a classical concert, the public hear the music but also can see the musicians, how they play or move, and which are their expressions and emotions. In EEG driven musical instrument, the musician must sit and stay immobile. We thought that adding a visual effect linked to the music could only improve the music. Therefore we studied different ways of showing the EEG. Finally we choose to present the signal projected on the brain cortex as explained in section V-B. When the musician is playing, every second, the recorded EEG are processed with the inverse solution approach and then averaged. An half sphere with the interpolation of the 361 solutions is projected on the screen (Fig. 8).

VII. CONCLUSION

During the workshop, two musical instruments based on biological signals were developed. One is based on EEG and the other on EMG. We have chosen the musical instrument approach rather than the sonification. Furthermore all the signals were used to spatialize and visualize the sound. We had not enough time to insert the heart sound and EOG into the composition.

One of the main achievements is the architecture we built. It enables the communication between any recording machine that can be linked to a network and a musical instrument. Since it is based on Matlab, any specific signal processing method can be easily implemented in the architecture. Furthermore the bridge built between Matlab and Max/MSP via Open Sound Control could be easily reused by other projects. Finally, we implemented basic and complex controls of the EEG.

The presented algorithm obtained 75% of accuracy for the classification of hand movements, that is the accuracy in controlling the synthesized musical sequence.

The present paper reflects the work of a four weeks workshop. However we will follow the study from several pathways. Our future research will aim at improving the instruments, from the signal recording part, up to the music synthesis. New methods for signal processing and classification should be tested to detect more tasks and to send more parameters to the instrument. Musician should be trained more adequately. Thus he will get a better control of the biological signal and by further practicing he will improve the mapping and begin to gain perfection on the instrument. In this context, musician's muscle activity should be vastly recorded (to give a bigger choice of choreography) as well as other biological signals (e.g. EOG used as a switch among instruments). EEG during the imagination of various musical themes can also be recorded and processed for the integration of an interacting perceptual component, that we did not take into consideration yet. Finally, the closer biological signal sonification will be of the musical instrument, the closer we will be of the dream.

REFERENCES

- [1] A. Tanaka, "Musical performance practice on sensor-based instruments," in *Trends in Gestural Control of Music*, M. M. Wanderley and M. Battier, Eds. IRCAM, 2000, pp. 389–406.
- [2] Y. Nagashima, "Bio-sensing systems and bio-feedback systems for interactive media arts," in *2003 Conference on New Interfaces for Musical Expression (NIME03)*, Montreal, Canada, 2003, pp. 48–53.
- [3] R. Knapp and A. Tanaka, "Multimodal interaction in music using the electromyogram and relative position sensing," in *2002 Conference on New Interfaces for Musical Expression (NIME-02)*, Dublin, Ireland, 2002, pp. 43–48.
- [4] Lusted, H. S. and Knapp, R. B., "Controlling computers with neural signals" in *Scientific American*, vol. 275, pp. 82–87, 1996.
- [5] J. R. Wolpaw, D. J. McFarland, G. W. Neat, and C. A. Forneris, "An EEG-based brain-computer interface for cursor control," *Electroencephalography and Clinical Neurophysiology*, vol. 78, pp. 252–259, 1991.
- [6] G. Pfurtscheller, C. Neuper, C. Guger, H. W., H. Ramoser, A. Schlögl, B. Obermaier, and M. Pregenzer, "Current trends in brain-computer interface (bci) research," *IEEE Transactions on Rehabilitation Engineering*, vol. 8, no. 2, pp. 216–219, jun 2000, comment.
- [7] J. R. Wolpaw, N. Birbaumer, D. J. McFarland, G. Pfurtscheller, and T. M. Vaughan, "Brain-computer interfaces for communication and control," *Clinical Neurophysiology*, vol. 113, pp. 767–791, 2002, invited review.
- [8] "BCI special issue," *IEEE Transactions on Biomedical Engineering*, vol. 51, 2004.
- [9] A. Brouse, "Petit guide de la musique des ondes cérébrales," *Horizon 0*, vol. 15, 2005.
- [10] E. Miranda and A. Brouse, "Toward direct brain-computer musical interface," in *2005 Conference on New Interfaces for Musical Expression (NIME05)*, 2005.
- [11] J. Berger, K. Lee, and W. Yeo, "Singing the mind listening," in *Proceedings of the 2001 International Conference on Auditory Display*, Espoo, Finland, 2001.
- [12] G. Potard and G. Shiemer, "Listening to the mind listening : sonification of the coherence matrix and power spectrum of eeg signals," in *Proceedings of the 2004 International Conference on Auditory Display*, Sydney, Australia, 2004.
- [13] J. Dribus, "The other ear : a musical sonification of eeg data," in *Proceedings of the 2004 International Conference on Auditory Display*, Sydney, Australia, 2004.
- [14] H. Berger, "Über das elektroencephalogramm des menschen," *Arch. f. Psychiat.*, vol. 87, pp. 527–570, 1929.
- [15] A. Lucier and S. Douglas, *Chambers*. Middletown: Wesleyan University Press, 1980.

- [16] D. Rosenboom, *Biofeedback and the Arts: Results of early experiments*, D. Rosenboom, Ed. Vancouver: Aesthetic Research Centre of Canada, 1976.
- [17] —, *Extended musical interface with the human nervous system*. Berkeley, CA: International Society for the Arts, Science and Technology, 1990.
- [18] M. Eaton, *Bio-Music: Biological feedback, experiential music system*. Kansas City: Orcus, 1971.
- [19] B. Knapp and H. Lusted, "Bioelectric controller for computer music," *Computer Music Journal*, vol. 14, pp. 42–47, 1990.
- [20] J. Vidal, "Towards direct brain-computer communication," *Annual review of Biophysics and Bioengineering*, pp. 157–180, 1973.
- [21] (2005, September) Mathworks. [Online]. Available: <http://www.mathworks.com/>
- [22] (2004, July) Open sound control. [Online]. Available: <http://www.cnmat.berkeley.edu/OpenSoundControl/>
- [23] (2004, July) Max/MSP. [Online]. Available: <http://www.cycling74.com/products/maxmsp.html>
- [24] DTI: Développement en traitement de l'information [Online]. Available: <http://www.dti-be.com/>
- [25] BIOPAC Systems Inc. [Online]. Available: <http://www.biopac.com/>
- [26] OpenEEG project. [Online]. Available: <http://openeeg.sourceforge.net/>
- [27] National Instruments Daqpad 6052e data acquisition card. [Online]. Available: <http://sine.ni.com/nips/cds/view/p/lang/en/nid/13893/>
- [28] D. Zicarelli, "An extensible real-time signal processing environment for max," in *Proceedings of the International Computer Music Conference*, Ann Arbor, Michigan, 1998.
- [29] Cycling'74. [Online]. Available: <http://www.cycling74.com/>
- [30] S. Mallat, *A wavelet tour of signal processing*. Academic Press, 1998.
- [31] Y. Wang, P. Berg, and M. Scherg, "Common spatial subspace decomposition applied to analysis of brain responses under multiple task conditions: a simulation study," *Clinical Neurophysiology*, vol. 110, pp. 604–614, 1999.
- [32] M. Cheng, W. Jia, X. Gao, S. Gao, and F. Yang, "Mu rhythm-based cursor control: an offline analysis," *Clinical Neurophysiology*, vol. 115, pp. 745–751, 2004.
- [33] P. Berg and M. Scherg, "A fast method for forward computation of multiple-shell spherical head models," *Electroencephalography and clinical Neurophysiology*, vol. 90, pp. 58–64, 1994.
- [34] J. C. Mosher, R. M. Leahy, and P. S. Lewis, "EEG and MEG: Forward solutions for inverse methods," *IEEE Transactions on Biomedical Engineering*, vol. 46, no. 3, pp. 245–259, March 1999.
- [35] S. Baillet, J. C. Mosher, and R. M. Leahy, "Electromagnetic brain mapping," *IEEE Signal processing magazine*, pp. 14–30, November 2001.
- [36] C. M. Michel, M. M. Murray, G. Lantz, S. Gonzalez, L. Spinelli, and R. Grave de Peralta, "EEG source imaging," *Clinical Neurophysiology*, vol. 115, pp. 2195–2222, 2004.
- [37] R. D. Pascual-Marqui, "Review of methods for solving the EEG inverse problem," *International Journal of Bioelectromagnetism*, vol. 1, no. 1, pp. 75–86, 1999.
- [38] B. Truax, "Time shifting of sampled sound with a real-time granulation technique," in *Proceedings of the 1990 International Computer Music Conference*, Glasgow, UK, 1990, pp. 104–107.
- [39] D. Trueman and L. Dubois. (2002) Percolate. [Online]. Available: <http://music.columbia.edu/PerColate>
- [40] L. Kessous and D. Arfib, "Bimanuality in alternate musical instruments," in *2003 Conference on New Interfaces for Musical Expression (NIME03)*, Montreal, Canada, 2003, pp. 140–145.

under the GNU/Linux Operating System. In our case, it worked effectively in sending messages to Macintosh computers running Max/MSP under MacOS X. The second approach used the pnet TCP/UDP/IP toolbox freely available from Mathworks. In this case, packets formatted according to the OSC protocol, were written to a network socket using the pnet command.

Example:

```
% head of the message
pnnet(udp,'write','/alpha');
% mandatory zero to finish the string
pnnet(udp,'write',uint8(0));
...
% comma to start the type tag
pnnet(udp,'write',',');
% number of float to write
pnnet(udp,'write','f');
...
% data to send
pnnet(udp,'write',single(data(i)),'intel');
```

This approach worked fine for most of our various computers running different operating systems, i.e. from Matlab on Linux or Windows to Max/MSP on Macintosh. However this did not work properly when we sent data to Max/MSP running on Windows due to endian problems. The toolbox has a byte swap function to accommodate for endianness and the correct one should be chosen, (see last command of the last line of the above example.) For more details on which endianness to choose see the pnet toolbox help.

APPENDIX I

OPEN SOUND CONTROL

To link Matlab and Max/MSP we used two approaches. The first one is based on a C++ library, liblo (<http://plugin.org.uk/liblo/>), which implements the OpenSoundControl and UDP protocols. The library is compiled as a Matlab plugin using the mex compiler. The file sendmat.c is an example of how to send a message from Matlab. All the functions of the OSC protocol should be accessible in this manner but only those in the example file were implemented. This has to date only been implemented

Multimodal Focus Attention Detection in an Augmented Driver Simulator

Alexandre Benoit, Laurent Bonnaud, Alice Caplier, Phillipe Ngo
Laboratoire des Images et des Signaux, Grenoble, France

Lionel Lawson, Daniela G. Trevisan

Communications and Remote Sensing Laboratory, Université catholique de Louvain, Belgium

Vjekoslav Levacic

Faculty of Electrical Engineering and Computing at University of Zagreb, Croatia

Céline Mancas

Faculté Polytechnique de Mons, Belgium

Guillaume Chanel

Université of Genève, Switzerland

Abstract— This project proposes to develop a driver simulator, which takes into account information about the user state of mind (level of attention, fatigue state, stress state). The user's state of mind analysis is based on video data and physiological signals. Facial movements such as eyes blinking, yawning, head rotations... are detected on video data: they are used in order to evaluate the fatigue and attention level of the driver. The user's electrocardiogram and galvanic skin response are recorded and analyzed in order to evaluate the stress level of the driver. A driver simulator software is modified in order to be able to appropriately react to these critical situations of fatigue and stress: some visual messages are sent to the driver, wheel vibrations are generated and the driver is supposed to react to the alertness messages. A flexible and efficient multi threaded server architecture is proposed to support multi messages sent by different modalities. Strategies for data fusion and fission are also provided. Some of these components are integrated within the first prototype of OpenInterface (the Multimodal Similar platform).

Index Terms— driver simulator, facial movements analysis, physiological signals, stress, attention level, data fusion, fission, OpenInterface.

I. INTRODUCTION

The main goal of this project is to use multimodal signals processing to provide an augmented user's interface for driving. The term augmented here can be understood as an attentive interface supporting the user interaction. So far at most basic level, the system should contain at least four components:

1. sensors for determining user state of mind;
2. an inference engine feature extractor to evaluate incoming sensor information

3. an adaptive user interface based on the results of step 2
4. an underlying computational architecture to integrate these components.

In fact a fully functioning system would have many more components, but the previous components are the most critical for inclusion in an augmented cognition system and they are covered in the project implementation.

Basically to provide such multimodal application, we address the following issues: which driver simulator to use? How to characterize a user's state of fatigue or stress? Which biological and/or physiological signals to take into account? What kind of alarm to send to the user? How to integrate all these pieces – data fusion and fission mechanism? Which software architecture is more appropriate to support such kind of integration?

A software architecture supporting real time processing is the first requirement of the project because the system has to be interactive. A distributed approach supporting multi thread server can address such needs.

The choice of the driver simulator has to take into account some features such as: open source software, “First person view”: (i.e. cockpit view with wheel) and dashboard, source code easy to modify and possible use of a vibration feedback wheel.

We consider two user's states to be detected: stress and fatigue. The detection of these states is based on video information and/or on biological information. From video data we extract relevant information to detect fatigue state while the biological signals provide data for stress detection. Physiological signals could be associated to video data in order to detect fatigue but in the context of the driver simulator used in this project, a real fatigue state is very difficult to obtain. It is possible to simulate fatigue on video data (by closing the eyes or by yawning for example). On the contrary, such kind of simulation is not possible on physiological signals.

This report, as well as the source code for the software developed during the project, is available online from the eINTERFACE'05 web site: www.enterface.net.

The third step is to define what kind of alarms to provide to the user. Textual messages and force feedback are considered to alert the user.

An other challenge of this project is to provide a concrete example in order to test OpenInterface, the multimodal Similar platform.

The rest of the paper is organized as follow: section II present the global architecture of the demonstrator, section III describes how we detect driver's hypo-vigilance states by the analysis of video data, section IV presents how to detect driver's stress states by the analysis of some biological signal, sections V and VI describe the data fusion and fission strategies and section VII gives details about the demonstrator implementation.

II. CONCEPTUAL ARCHITECTURE

The diagram of Figure 1 presents the conceptual architecture of our attentive driver simulator. We propose a distributed approach to integrate our components. On one PC under Linux we have integrated all video data based detection and analysis as well as the fusion and fission components. An other PC under Windows is used to run the driver simulator and a third PC is used for biological signals acquisition and analysis. Communication between all the PCs is done exchanging XML messages. For that the Dialog Controller included in the driver Simulator software should be able to receive multi messages (i.e. from biological signals station and from video based station). In this case a multi thread server approach is developed and included in the driver simulator.

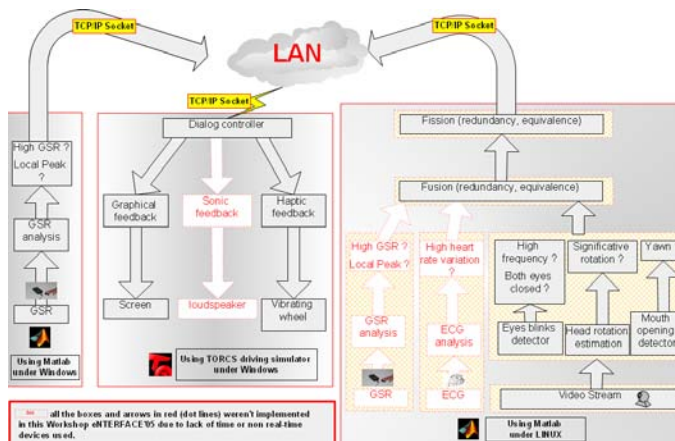


Figure 1: Overview of the system architecture

III. HYPO-VIGILANCE DETECTION BASED ON VIDEO DATA

The state of hypo-vigilance (either related to fatigue or inattention) is detected by the analysis of video data. The required sensor is a camera facing the driver. In this project, three indices are considered as hypo-vigilance signs: yawning, head rotations and eyes closing for more than 1s.

A. Face detection

Face detection is the first and maybe the most crucial step of the image processing phase. The face detector should be robust (no error in face localization) and should work in real time. The chosen face detector is the free toolbox MPT [5]. This face detector extracts a square-bounding box around each face in the processed image. Face detection is done for each image of the sequence without any face tracking. The advantage is that the head is not lost because of tracking error propagation. The main drawback is the decrease of the frame rate even though MPT works nearly in real time for pictures of size (320x200 pixels), which is not the case of other face detectors such as OpenCV [13] for example.

Whichever face detector you use, the extracted face bounding box is not exactly the same from frame to frame so that we use a temporal median filter with temporal adaptive position mean to make the spatial localization of the face temporally stable (note that the size of the bounding box of the face is supposed to be constant during an experiment: in a car the driver face distance w.r.t the camera is stable if the driver stays on his seat).

B. Head motion analysis

Once a bounding box around the driver face has been detected, head motion such as head rotations, eyes closing and yawning are detected by using an algorithm working in a way close to the human visual system. In a first step, a filter coming from the modeling of the human retina is applied. This filter enhances moving contours and cancel static ones. In a second step, the FFT of the filtered image is computed in the log polar domain as a modeling of the primary visual cortex. Figure 2 gives a general overview of the algorithm.

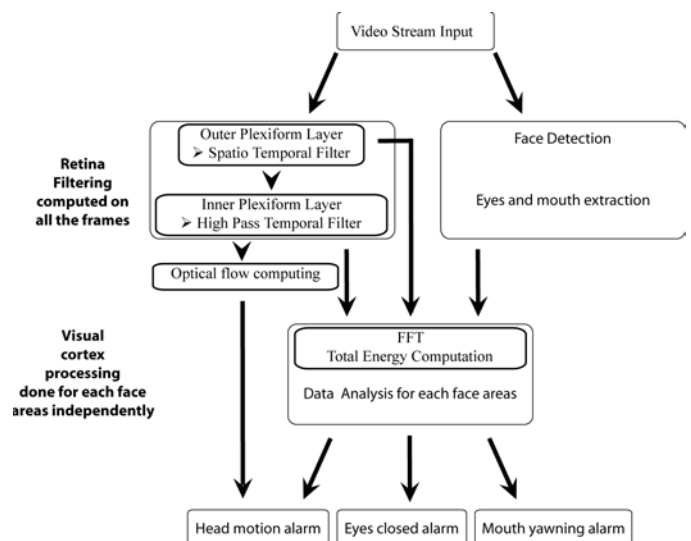


Figure 2: Algorithm for hypo-vigilance features extraction from video data

The first step consists in an efficient prefiltering [1]: the retina OPL (Outer Plexiform Layer) that enhances all contours by attenuating spatio-temporal noise, correcting luminance and

whitening the spectrum (see Figure 3) . The IPL filter (Inner Plexiform Layer) [1] removes the static contours and extracts moving ones.

The second step consists in a frequency analysis of the spectrum of the OPL and IPL filters outputs in each region of interest of the face: global head, eyes and mouth (see section C for the description of eyes and mouth region of interest extraction).

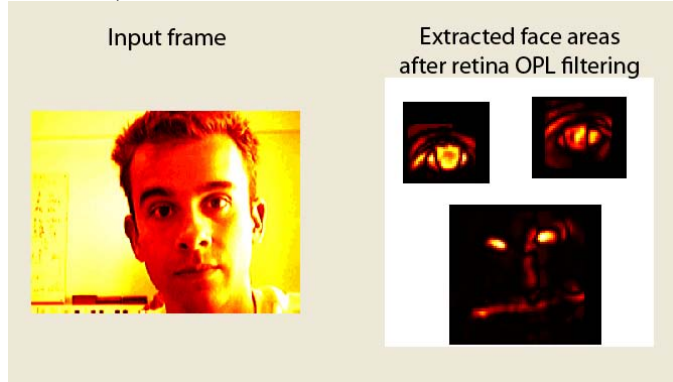


Figure 3 : OPL filtering results

In order to estimate the rigid head rotations [3], the proposed method analyses the spectrum of the IPL filter output in the log polar domain. It detects head motion events and is able to extract its orientation. Cortical optical flow filters [2] are oriented filters that compute the optical flow of the global head to extract the motion direction. Optical flow is computed only when motion is detected.

For the detection of yawning or eyes closing, three identical processes are done independently [4]. On each region of interest (each eye and the mouth), a spectrum analysis of the OPL and IPL filters output is done for motion event detection: we are looking for vertical motion related to eyes closing or to yawning.

C. Eyes and mouth detection

The mouth can be easily extracted in the lower half of the detected bounding box of the head. The detection algorithm will work even if the mouth is not perfectly centered in the area because we analyze the spectrum energy instead of spatial features, which is more robust. Moreover, there are no disturbing contours in that area that could generate false detections.

Concerning the eyes, the spectrum analysis in the region of interest is accurate only if each eye is correctly localized. Indeed around the eyes, several vertical or horizontal contours can generate false detection (hair boundary for example).

The MPT toolbox proposes an eye detector but it requires too much computing time so that it is not compliant with real time constraint. We use another solution: eye region is supposed to be the area in which there is the most energized contours. To do so, assuming that the eyes are localized in the 2 upper quarters of the detected face, we use the retina output. The retina output gives the contours in these areas and due to the fact that the eye region (containing iris and eyelid) is the only area in which there are horizontal and vertical contours, the eye detection can be achieved easily. We use two oriented low pass filters: one horizontal low pass filter and a vertical low

pass filter and we multiply their response. The maximum of the result is obtained in the area in which there are the most horizontal and vertical contours that is an eye region. To make the eye areas temporally stable, their position is smoothed from frame to frame using adaptive mean positions. This eye detection takes about 6 operations per pixel for each search area (i.e. each upper quarter of the face bounding box).

Figure 4 gives an example of the input picture of the eye detector; bright areas are the most important contours. Figure 5 shows the output of the horizontal vertical filters.

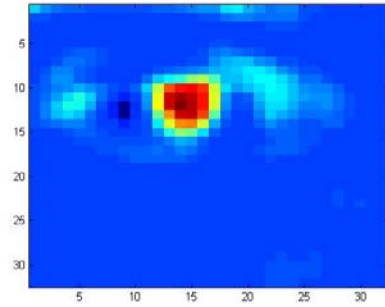


Figure 4: Input picture for eye detection : one of the 2 upper quarters of the face-bounding box

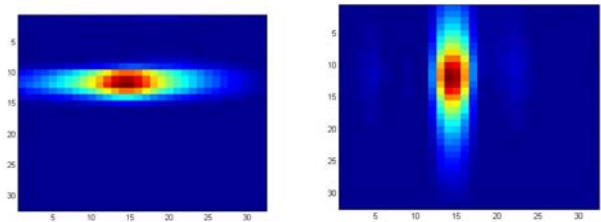


Figure 5: Output of the vertical and horizontal low pass filters, both filters report maximum amplitude on the eye center.

D. Hypo-vigilance alarms generation

- We generate an alarm when both eyes are closed longer than a specific time period (1 second for example).
- We detect mouth yawning: when a yawn occurs, the mouth is wide open, then, this generates a very high-energy increase on the spectrum that can be easily extracted.
- The global head motion events are detected with the global head spectrum analysis. We only extract the fact that a head motion has occurred. The proposed algorithms are able to extract the motion direction with the cortical optical flow algorithm, but it is not yet integrated in the fusion system.

E. Fusion strategy

After the video analysis, Boolean information about yawning or not, about eyes closing or not and about head moving or not are available. A very simple and easy to compute fusion strategy based on the three index is proposed:


```

if head motion is detected
    send an alarm to the user
    hypo-vigilance value=100
else
    if both eyes are closed during 1s
        send an alarm to the user
        hypo-vigilance value = 50
    if the driver is yawning
        send an alarm to the user
        hypo-vigilance value = 50+hypovigilance value
end

```

The variable hypo-vigilance associated to each index is set to 50 or 100. The highest the value, the highest the hypo-vigilance.

Note that in this very simple fusion strategy, information about head motion kind of rotation is not taken into account. A more sophisticated fusion strategy has been tested and is described in section V.

IV. STRESS DETECTION BASED ON BIOLOGICAL SIGNALS ANALYSIS

Physiological signals are used in order to detect stress situation. ECG (Electrocardiogram) and GSR (Galvanic Skin Response) are announced by literature as very promising to detect driver stress in real situations [11, 12]. In a stressful time, the GSR signal and the heart rate signal (extracted from the ECG) are supposed to increase. Two different experiments have been considered; they aim at detecting either driver stress over a long time period or punctual driver stress.

In this experiment, we use the Biopac system MP30B-CE for ECG and GSR acquisition.



Figure 6: On the left, ECG devices and on the right, GSR device

The main drawback of the data acquisition system is that for the moment, on line analysis is not possible. For that reason, the study on biological signal for stress detection has not been implemented in the final demonstrator.

A. ECG signals analysis

1) Prefiltering

Since we are analysing the stress state of a driver, the ECG can be disturbed by several muscle artefacts that generally come from hands or arms movements (see Figure 7). This is why it is necessary to pre-filter signals.



Figure 7: Driver with electrodes on the wrists for ECG measure

The pre-filtering is based on the characteristics of the ECG peaks: we observed that these peaks contain energy in the frequency band 10-35 Hz. As we do not need other components of the signal we choose to band-pass the signal using this interval and a Butterworth IIR filter with 8 coefficients. Figure 8 shows the original signal and the results after filtering.

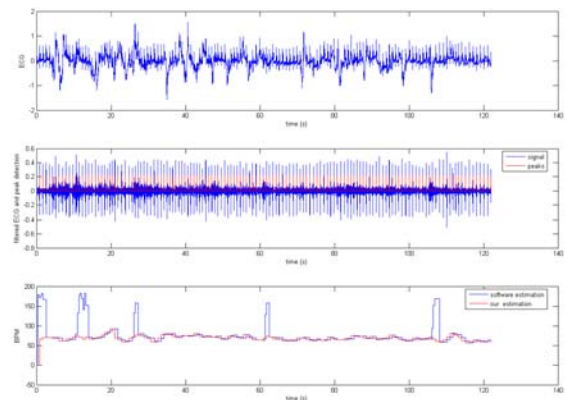


Figure 8: ECG before (first line) and after pre-filtering (second line); computed heart rate from ECG (third line, red signal)

2) Heart rate computation

The first thing to do, before computing heart rate, is to identify peaks in the filtered signal. For this, we use a reference record (ECG when the subject is supposed to be relaxed) as a baseline to identify the general height of the peaks depending on the subject. We define the general height of peaks as one third (chosen empirically) of the maximum value. In order to improve the peak detection, we also use a priori information: we consider that the heart rate cannot exceed 180 BPM (Beats Per Minute). If two peaks are too close, so that they do not validate this assumption, we keep only the one with the maximum value.

Finally, the heart rate is computed by evaluating the number n of samples between two peaks and by using this simple formula:

$$HR = 60 / (n * (1/f_e))$$

Where f_e is the sampling rate.

3) Stress level assessments

In case of unexpected or stressful events, the heart rate does not increase as generally assumed, but one can observe a raise in its variation. In order to determine this variation, we use the absolute value of the first derivative of the signal. After smoothing the result by a Gaussian filter we obtain what we call the stress level (see Figure 9).

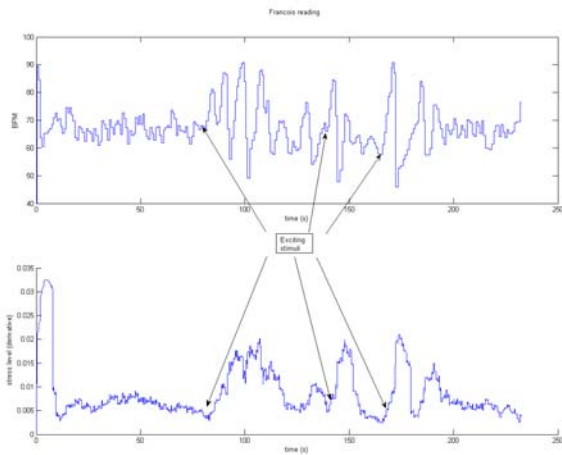


Figure 9 : stress level (bottom) computed from heart rate (top) for exciting stimuli

B. GSR signal analysis

Due to the chosen sampling rate (200 samples per second) and the apparatus, some artefacts occur in the initial signals and a filtering is also required. After trying several smoothing filters and because of the large variability in the conductivity of each user, we opt for a multiscale median filtering. Four successive filters are used with a decreasing window size (100, 50, 30 and 20 samples).

1) Global stress detection

We can easily measure the minimum and maximum of the user's GSR level. By normalizing the signal to analyze with these values as usual:

$$\frac{GSR_level_to_analyse - \min(GSR_rest)}{\max(GSR_rest) - \min(GSR_rest)}$$

where GSR_rest correspond to the values of the GSR when the user is supposed to be relaxed.

We can then define a score to know the global stress level. Global stress can occur after a very difficult day of work or when the traffic jam is increasing for example. Global stress is related to slow but constant increase of the GSR. The global stress level is used to know the initial state of the driver or with a sliding window, to know the global state of the driver in a certain amount of time.

2) Local stress detection

Local stress is supposed to be related to punctual and unforeseen events as it could occur on roads such as a

pedestrian crossing and so on. Local stress detection can be modelled as high peaks in GSR signals. GSR signals have the property to react quite quickly to an event but to have a decreasing response to go back to a calm situation very slowly.

We used this particular property as a priori information in our algorithm.

First of all, we detect local maxima using the watershed algorithm. Local maxima correspond to watershed pixels. Once maxima are detected, we keep only those, which have a difference of 1 unit with the previous maximum. This threshold of 1 unit is based on the correlation of punctual events, precisely recorded, and GSR signals. Then, we remove all maxima, which are too close to each other by keeping only the highest one. This rule is based on the assumption that if two maxima are too close to each other (inferior to 5 seconds), they belong to the same event. Figure 10 presents an example of GSR local increasing detection: each peak on the bottom curve corresponds to a stress alert.

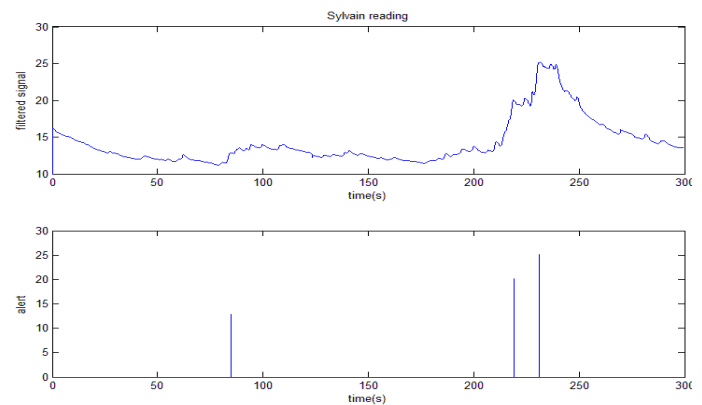


Figure 10: GSR record and punctual stress event detection

V. FUSION STRATEGY

In this section, we describe and test a data fusion based on Bayesian Network. It is used for the purpose of hypo-vigilance detection but it also represents a global fusion method for the integration of additional information in the detection process. Note that this fusion process is not integrated in the final demonstrator for the moment due to the lack of significant data. Both fusion strategies are implemented in the demonstrator but for the moment, only the simplest one described in III.D. is used by default for computational cost reduction.

Human fatigue generation is a very complicated process. Several uncertainties may be present in this process. First, fatigue is not observable and it can only be inferred from the available information. In fact, fatigue can be regarded as the result of many contextual variables such as working environments, health and sleep history. Also, it is the cause of many symptoms, e.g. the visual cues, such as irregular eyelid movements, yawning and frequent head tilts. Second, human's visual characteristics vary significantly with age, height, health and shape of face. To effectively monitor fatigue, a system that integrates evidences from multiple sources into

one representative format is needed. Naturally, a Bayesian Networks (BN) model is a good option to deal with such an issue.

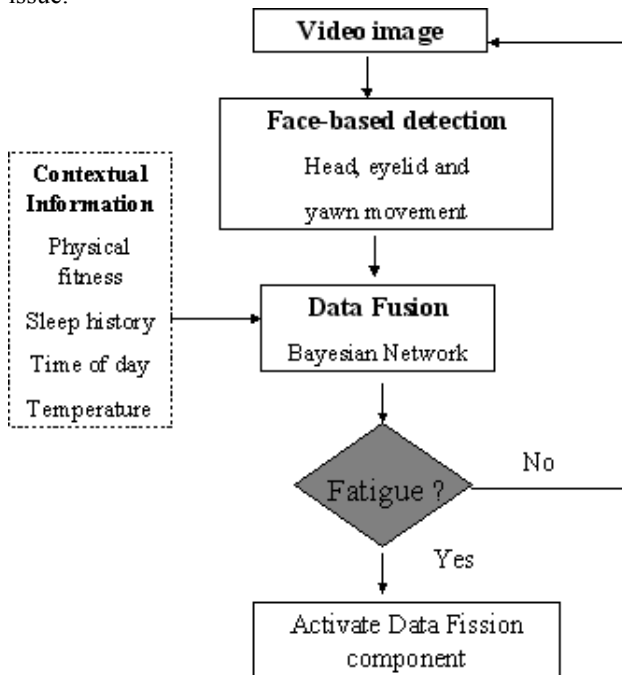


Figure 11: Fusion strategy based on a Bayesian Network

A BN provides a mechanism for graphical representation of uncertain knowledge and for inferring high-level activities from the observed data. Specifically, a BN consists of nodes and arcs connected together forming a directed acyclic graph. Each node can be viewed as a domain variable that can take a set of discrete values or a continuous value. An arc represents a probabilistic dependency between the parent node and the child node.

Some contextual information such as temperature, time of day, sleep history, etc can be used to build a prior probability for the fatigue node. For that we use the parameters proposed in [7]. For the face data fusion we have considered a very preliminary version where the network evidences change when: eyes closed more than 1 sec; yawning occurs; down head motion are detected simultaneously or not. As result we got the level of fatigue, which is sent to the data fission component.

VI. FISSION STRATEGY

Data fission duty is to collect the data from data fusion and to generate an alert XML message that is sent to the driver simulator. Data fission function is called at the rate the driver state detection is progressing. Generated messages are in XML format. We decided for XML because it is extendable and messages are sent only when the driver state changes. Driver state may be defined by a fatigue value (either coming from the Bayesian Network result or from the simple fusion process) that is an output variable of data fusion. For example, we can set the range of values for fatigue level that determine the driver state. For those range of values we can define different screen messages and wheel shaking power. Table 1

and Table 2 present the fusion strategy for the simple method and for the Bayesian network based method respectively.

Fatigue range	50	50	100
Message	Open the eyes	Yawning: be careful	Stop moving the head
Shaking power	'100'	'100'	'100'

Table 1: fission strategy with the simple fusion process

Fatigue range	[0,33]	[33,66]	[66,100]
Message	"	'Tired'	'Asleep'
Message color	"	'Green'	'Red'
Shaking power	'0'	'0'	'100'

Table 2: fission strategy with the BN based method

Data fission only creates the message if the driver state has changed and is different than the previous driver state. If the user state is the same as in previous call, data fission generates 'NOT_CHANGED' message. In that way the XML message does not need to be sent to the driver simulator after each call of the data fission function.

Once the alert message has been sent, the driver is supposed to acknowledge to the system that the message has been understood. For example, in the case of the simple fusion process, each time an alert is detected, wheel vibrations are triggered. The driver has to stop these vibrations by pushing a button. The reaction time is also recorded, this time being correlated with the hypo-vigilance or fatigue user state.

VII. DEMONSTRATOR

A. Overview of the global system

The developed demonstrator is made of

- 2 PCs: one under Windows for the driver simulator and one under Linux for hypo-vigilance states detection
- 1 SONY digital camera
- 1 LOGITECH force feed back wheel
- 1 projection screen
- 1 video-projector
- 2 loudspeakers

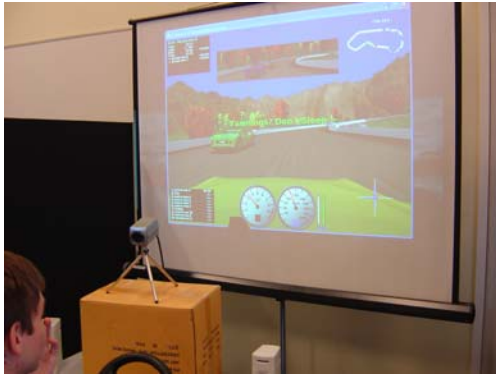


Figure 12: Global views of the demonstrator

On the used computer (Pentium 4 2.4Ghz), the frame rate is about 5 frames per second but it could be increased up to 8 frames per second thanks to some MPT optimization.

B. Driver Simulator

Around ten driver simulators have been studied. The chosen driver simulator is TORCS [9] because it is a well architected GPL program with well structured source code and a well designed user interface.

This simulator is working under Linux and windows platforms. The main sources are written in C++ with the OpenGL library. The graphics quality of the simulator is correct and it has a first person view. Figure 13 presents an illustration of TORCS simulator.



Figure 13: Torcs driver simulator illustration

We integrate an interaction from the Data Analysis Kernel to our driving Simulator.

The main work consisted in

- Allowing a Text Message to be displayed within the game graphical interface.
- Creating a multi-threaded Server within the application whose purpose is accepting different clients connexions.
- Integrating a force Feedback wheel in order to warn the user with an other modality than the visual one.
- Allowing the user to make a feedback on the message displayed by stopping it.
- Parsing XML messages from the multimodal analysis of the driver. Indeed, it is possible to change the color, the string of the sent message and the feedback power.

C. Implementation of hypo-vigilance detection

Due to the fact that ECG and GSR signals cannot be processed on line with the data acquisition station we used, the detection of stress state has not been implemented in real time. Only the detection of hypo-vigilance state based on video data is available at the moment.

1) Face detection algorithm modifications

For face detection, we use the Matlab implementation of *mpiSearch* function belonging to the MPT library, which receives a RGB or Gray level frame as input. Outputs of the function are the bounding box coordinates of the detected face.

Due to the relative slowness of the *mismatch* function developed under Matlab a fine study of the algorithm has been done in order to increase the computational rate. We managed to figure out how the algorithm behaves in dynamic environment when the video is acquired with the help of DirectInput library. While streaming, *mpiSearch* uses a special object that caches the detected face-bounding box.

The trick is to modify the *mpiSearch* Mex function and to put this object as a global DLL variable. Global DLL variables are preserved in Matlab memory space after the DLL is first accessed by Matlab. In that way after each Matlab call of the *mpisearch* function caching state is preserved.

With using the Microsoft VS.NET 2003, we additionally increase the performances by compiling the code for new Pentium 4 generation of processors.

With all these modifications, we achieve about 2.7 times speed increase.

In order to increase the frame rate again, some of time-consuming parts of the *mpisearch* code may be written in assembler language. This is beyond of scope of the project.

D. Alert message generation

1) The Display Changes

Three different messages depending on the index of hypo-vigilance can be displayed on the first view of the driver simulator (see Table 1 for the different considered messages and Figure 14 for an example of message incrustation during the game). In order to show the message, we need to change all the graphical classes within the source code. The communication between the main program and all the libraries is created by using some global variables and also by creating new links between several libraries.

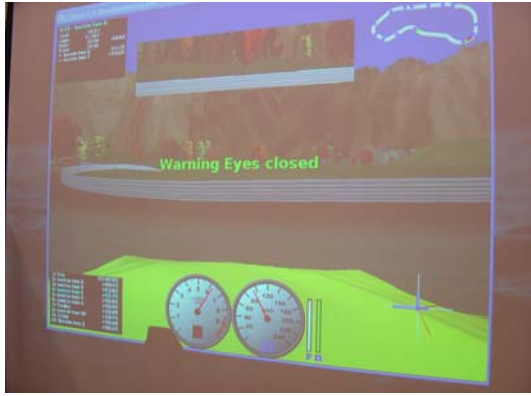


Figure 14: alert message in case of long eyes closing

2) Force Feedback implementation

Force Feedback is used to make the wheel shaking when hypo-vigilance is detected. Shaking power is defined by XML message of data fission. Shaking becomes stronger and gradually reaches its maximum value. Force Feedback uses the DirectInput library, a part of the Microsoft DirectX SDK. The library is based on "The Force Feedback Direct Input Library (DIL)" made by Bryan Warren and Alex Koch. This library can be loaded at [8]. In our project, the library has been altered from functions to class. In class we can set the time period that shaking needs to reach the maximum vibration time, vibration activation threshold and to modify the shaking of the wheel dynamically. We also use the class to check whether the button is pressed, and if pressed stop the wheel shaking.

3) Message parsing and controlling the input devices of driver simulator

After the XML message arrives through the socket connection it is parsed. We use Microsoft XML parser to parse the message. You can download the MSXML parser from Microsoft web site. After the parsing, controller class activates the screen display message or starts the wheel shaking.

4) The Server Side:

We choose to implement a Server side for the interaction between the multimodal devices and the user. The network protocol used is TCP/IP. We implement this socket by using threads. Those threads access global variables under mutual exclusion. We use a "GPL" library called Openthreads for this implementation.

E. Openinterface integration

1) OpenInterface: a short presentation

OpenInterface is the Similar Software Platform that includes software components dedicated to multimodal interaction and multimodal data fusion. OpenInterface integrates results from the Human-Computer Interaction (HCI) community as well as from the Signal Processing community.

Each component is registered into OpenInterface Platform using the Component Interface Description Language (CIDL described in XML). The registered components properties are

retrieved by the Graphic Editor (Java). Using the editor the user can edit the components properties and compose the execution pipeline of the multimodal application. This execution pipeline is sent to the OpenInterface Kernel (C/C++) to run the application. The OpenInterface Tutorial can be found on the Similar web site [10].

2) OpenInterface implementation of the demonstrator

Currently, OpenInterface is in its early development stage and this driver simulator project is used as a test bed for the working prototype. The latter provides several services and also has limitations.

The services provided by this first OpenInterface prototype are:

- Interface Description Language for the specification of reusable software code.
- Seamless integration of reusable heterogeneous software (C/C++, Java, Matlab).

Some limitations are:

- The description of a software interface is currently not automated and has to be written by hand. This can be cumbersome as the language syntax is very strict and the validity of a description is not heavily enforced besides DTD checking. This lack of robustness might lead to an inappropriate binding and therefore to unexplained application crash.
- At the beginning of the project, it was impossible to perform two ways communication with any Matlab script. The latter could only be called by OpenInterface component.

Several issues have been encountered during the integration phase:

- First of all as the current prototype is only running under Linux, we had to reduce the set of software to those that could be run under Linux. It has been decided that the communication with non-compliant Linux software will be done through network communication. Thus, only the following software code have been integrated into OpenInterface: firewire camera driver, face detection, head motion analysis, fusion for hypo-vigilance detection, fission strategy for alerting the user and a Java GUI to start the application. Figure 15 shows how these components are connected into OpenInterface.
- After deciding which software to use as component into OpenInterface, the main problem we had to face has been to make those components reusable. Indeed, in their first version, the software was designed for a particular goal and could not be reused by a third-part (e.g. OpenInterface) as tools. Thus, we redesigned the components interaction interfaces. To avoid this redesigning step in the future, modular programming habit should be enforced from the beginning among components programmer.
- From this step we have pulled out that a two-way communication with Matlab script is mandatory to

prevent heavy code rewriting. Indeed the camera driver has to be called by the face detection component, not the contrary. Therefore, due to the Matlab Engine API limitations, the only way to allow Matlab script to interact with component already running into OpenInterface is to perform communication through UNIX socket. Of course this is transparent to the user.

- The third integration step was the description, in CIDL, of all components interface. This step was straightforward and we did not find any difficulties in expressing the interfaces in the integration platform's CIDL.
- Finally we started the integration of the components. It is performed in a gradual way so that we were able to test each component inside OpenInterface and point out the incompatibility. Some problems arose when we put two Matlab components in the same execution pipeline. It turned out that due to another undocumented Matlab Engine API limitation, we could not call another m-script while another is running. The lack of time directed us to fix the problem by having one Matlab engine instance per Matlab script taking part in a pipeline. One would think that this solution would drastically slow down the computer but that did not happen. A Matlab process actually uses resource proportional to the computation done by the running script.

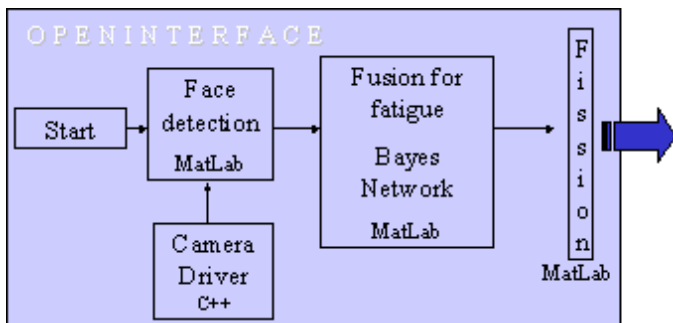


Figure 15: Openinterface components coming from the project

As a conclusion of this integration phase using OpenInterface platform, we can say that thanks to this first project integration a lot of feedback has been collected to improve the current prototype of the OpenInterface tool in order to provide another improved version as soon as possible. The future engineering work on OpenInterface will be narrowed to improve the integration of Matlab components and also to provide a user with a friendly integration interface. This would allow distributing the platform and letting people test it on their own.

VIII. FUTURE WORKS AND CONCLUSION

During the project, we have developed an augmented driver simulator based on video analysis for driver's attention controlling. First promising studies about physiological data have to be improved and integrated in the global system. This

will induce the development of an appropriate data fusion method in order to control both the driver's attention level and the driver's stress.

Once the driver has been alerted, it will be necessary to perform some specific tests in order to control that driver's stress or fatigue has actually decreased.

For the moment, the global system is running almost 10 frames per second. It will be necessary to optimize video data analysis algorithms in order to speed up the frame rate.

ACKNOWLEDGMENT

This work was supported by the Similar network of excellence [10]

REFERENCES

- [1] Beaudot W., "The neural information processing in the vertebrate retina: A melting pot of ideas for artificial vision", PhD Thesis in Computer Science, INPG (France) december 1994.
- [2] Torralba A. B., Herault J. (1999). "An efficient neuromorphic analog network for motion estimation." IEEE Transactions on Circuits and Systems-I: Special Issue on Bio-Inspired Processors and CNNs for Vision. Vol 46, No. 2, February 1999.
- [3] Benoit A., Caplier A. "Head nods analysis : interpretation of non verbal communication gestures " IEEE, ICIP 2005, Genova, Italy
- [4] Benoit A., Caplier A. "Hypovigilance Analysis: Open or Closed Eye or Mouth ? Blinking or Yawning Frequency ?" IEEE, AVSS 2005, Como, Italy
- [5] Machine Perception Toolbox (MPT) <http://mplab.ucsd.edu/grants/project1/free-software/MPTWebSite/API/>.
- [6] Bayes Net Toolbox for MatLab [<http://www.cs.ubc.ca/~murphyk/Software/BNT/bnt.html>].
- [7] [Qiang Ji, Zhiwei Zhu and Peilin Lan, Real-Time Nonintrusive Monitoring and Prediction of Driver Fatigue, IEEE Transactions on Vehicular Technology, Vol. 53, No. 4, July, 2004, p1052-1068].
- [8] Force Feedback Direct Input Library http://courses.washington.edu/css450/Fall2003/web_contents/direct_input_lib/DirectInput.html
- [9] TORCS Driver Simulator: <http://torcs.sourceforge.net/>
- [10] Similar Network of Excellence: www.similar.cc
- [11] S. K. Lal, A. Craig, "Driver fatigue: electroencephalography and psychological assessment", Psychophysiology, 39, 3, May 2002, 313-21.
- [12] J. Healey, J. Seger, R. Picard, "Quantifying driver stress: developing a system for collecting and processing bio-metric signals in natural situation", MIT technical report n°483.
- [13] OpenCV : www.intel.com/technology/computing/opencv

Alexandre Benoit was born in 1980 in France. He graduated from Institut National Polytechnique de Grenoble (INPG). His PhD subject concerns head motion analysis. His work is based on the human visual perception system. He teaches signal processing to engineering students. He prepares his PhD from the INPG, his thesis started in october 2003 at the Laboratoire des Images et des Signaux (LIS) in Grenoble.

Laurent Bonnaud was born in 1970. He graduated from the École Centrale de Paris (ECP) in 1993. He obtained his PhD from IRISA and the Université de Rennes-1 in 1998. Since 1999 he is teaching at the Université Pierre-Mendès-France (UPMF) in Grenoble and is a permanent researcher at the Laboratoire des Images et des Signaux (LIS) in Grenoble. His research interests include segmentation and tracking, human motion and gestures analysis and interpretation.

Alice Caplier was born in 1968. She graduated from the École Nationale Supérieure des Ingénieurs Électriciens de Grenoble (ENSIEG) of the Institut National Polytechnique de Grenoble (INPG), France, in 1991. She obtained

her Master's degree in Signal, Image, Speech Processing and Telecommunications from the INPG in 1992 and her PhD from the INPG in 1995. Since 1997, she is teaching at the École Nationale Supérieure d'Électronique et de Radioélectricité de Grenoble (ENSERG) of the INPG and is a permanent researcher at the Laboratoire des Images et des Signaux (LIS) in Grenoble. Her interest is on human motion analysis and interpretation. More precisely, she is working on the recognition of facial gestures (facial expressions and head motion) and the recognition of human postures.

Guillaume Chanel was born in 1978 in Switzerland. He obtains both his engineering diploma in computing and robotics and his master degree in automatics during the 2002 year. He is currently a PhD student at the Computer Vision and Multimedia Laboratory (CVML) of the University of Geneva. His research interest is to detect emotional states from recordings of EEGs and other physiological signals in order to improve human computer interactions.

Lionel Lawson was born in 1982 in Bénin. He graduated from the Engineering School of Université Catholique de Louvain (UCL) and obtained his Master degree in Computer Science and Engineering in 2004. He is currently working at the Communication and Remote Sensing Laboratory (TELE) on the development of OpenInterface, an open source component-oriented integration platform.

Vjekoslav Levacic was born in 1981 in a small but pleasant city at north of Croatia called Cakovec. He ended two high schools, gymnasium and classical music high school. In 2000 he became a student in Faculty of Electrical Engineering and Computing in University of Zagreb. He has worked on various projects including building the enterprise systems and web applications.

Céline Mancas-Thillou holds two Master degrees, in Audiovisual Systems and Networks Engineering (ESIGETEL, 2002) and in Applied Sciences (FPMS, 2004). She is working for the TCTS lab since January 2003 and is pursuing a PhD in Applied Sciences since March 2004. Her research deals with text extraction, segmentation and degraded character recognition in SYPOLE project. She has been a visiting PhD student at the University of Bristol for 3 months in 2005 to work on Super Resolution Text for an embedded application.

Phillipe Ngo was born in 1980 in France. He is still undergraduated from the UTBM (university of technology of Belfort Montbéliard (France)). His major is computer science with specialization in Picture, interaction and virtual reality. He is currently working for the LIS (INPG Grenoble France) for its last internship of computer science. His work is focused on human and computer interactions within a virtual reality environment.

Daniela G. Trevisan was born in Santa Maria, Brazil, on 1974. She graduated in Informatic at the University Federal of Santa Maria, Brazil ([UFMS](#)) in 1997. She obtained Master degree in Computer Science from University Federal of Rio Grande do Sul, Brazil ([UFRGS](#)) in 2000. Currently she is PhD student at the Université Catholique de Louvain ([UCL](#)) at the Communication and Remote Sensing Laboratory ([TELE](#)) and she is also member of the Belgium Computer-Human Interaction Laboratory ([BCHI](#)). Her research topics are focused on human-computer interaction (HCI) field such as modelling multimodal interfaces, model based-approach, augmented and mixed reality and multimodal interfaces for image-guided surgery.

GMM-Based Multimodal Biometric Verification

Yannis Stylianou, Yannis Pantazis, Felipe Calderero, Pedro Larroy, Francois Severin,
Sascha Schimke, Rolando Bonal, Federico Matta, and Athanasios Valsamakis.

Abstract—In this work, we describe how biometric data can be used for person identification and verification. We rely on three categories of traits, that is speech, signature, and face. These distinguishing features or characteristics of a person, on their own, do not provide satisfactory results using well-known techniques. This is the case especially when the number of enrolled persons is large. For this reason, we develop techniques for making good use of all the three traits. In particular, we choose to follow late fusion of the scores of each single trait. The results of these techniques are quite better than using only one trait. Another goal of this work is the creation of a high quality multilingual database with video, audio, and signatures from forty seven persons.

Index Terms—biometrics, speaker recognition, on-line signature authentication, eigenfaces, fusion, multilingual database.

I. INTRODUCTION

OVER the past years, the need for secure transactions using biometric data has attracted a lot of attention. Knowledge-based techniques such as passwords suffer from various shortcomings as they can be forgotten or stolen. Biometric-based features promise easier interaction and potentially high security level.

The use of only one trait for person identification has been proved that is not enough for real life applications such as banking access. This problem is more evident when the number of enrolled persons is increasing. To meet real life applications demands, it is required to take advantage of not only one trait but of two or potentially three. In the current work we decided to make use of three easily acquired traits, that is speech, signature, and face. For this reason and in order to test our algorithms we have created a database of 47 persons. The database contains high quality video of approximately 4 min, speech and signature data of each person.

Algorithms that use biometrics for person identification/verification rely on two categories of traits: physiological and behavioral. Speech and signature can be put under the category of behavioral traits. In a more theoretical context, behavioral traits can be thought of as being different realizations of a stochastic (random) process. These kind of traits have the advantage that they are not easily copied. Physiological traits are constant for each person, for example fingerprint face and iris. These kind of features can also be use for identification purposes. In general, physiological features provide better

identification results but they suffer from various shortcomings as they can be duplicated.

For the current work, we have decided to make use of both behavioral and physiological traits. This is because we believe that a combination of both will provide better identification results therefore higher security for a potential application. The specific biometrics we use are speech and signature as behavioral traits and face as physiological trait. This particular decision has initiated from the fact that these kind of traits can be easily obtained by prevalent devices such as PDAs or mobile phones.

The whole system is divided in three agents (subsystems) one for every trait. An important point that must be addressed is the fusion of the results from the different subsystems. The procedure we follow here is by adding the different likelihoods from the three different agents (speech, signature and face agent) and picking the largest one.

The remainder of this report is organized as follows. Section II describes the different biometric traits, we used. Section III describes the fusion procedure in detail. In section IV, details about multimodal database are given. Section V shows the results of each agent and of the fusion. Finally, future directions are given in section VI.

II. BIOMETRIC TRAITS

A. Speech

Based on results of previous studies for automatic speaker recognition systems, we have used Mel-cepstral features. Which are one the most successful feature representations in speech recognition tasks.

The feature extraction consists of the following steps. Every 10 ms the speech signal is multiplied by a Hamming window $w[n]$ with a duration of 20 ms to produce a short time speech segment $x[n]$. The discrete Fourier spectrum is obtained via a fast Fourier transform from which the magnitude squared spectrum is computed. The magnitude spectrum $X[n]$ is put through a bank of filters. The filter bank used simulates critical band filtering with a set of triangular bandpass filters. The critical band warping is done following an approximation to the Mel-frequency scale which is linear up to 1000 Hz and logarithmic above 1000 Hz. The center frequency of the triangular filters follow a uniform 100 Hz Mel-scale spacing and the bandwidths are set so the lower and upper pass-band frequencies of a filter lie on the center frequencies of the adjacent filters, giving equal bandwidths on the Mel-scale but increasing bandwidths on the linear scale. The Mel-scale cepstral coefficients are computed from the filter bank outputs. The first coefficient $c[0]$ reflects the average log energy in the speech frame and is discarded as a form of amplitude normalization.

This report, as well as the source code for the software developed during the project, is available on-line from the eNTERFACE'05 web site: www.enterface.net.

This research was partly funded by SIMILAR, the European Network of Excellence on Multimodal Interfaces, during the eNTERFACE05 Workshop in Mons, Belgium.

B. Face

In our daily life, one of the most important and human-friendly biometrics to identify people is face recognition. Almost all recognition systems including human actors incorporate this modality, based on photographs or video sequences. For more than 20 years, understanding and developing face recognition systems has become a challenge able to seduce people from a wide range of research areas, from pattern recognition and computer vision to cognitive and perception sciences.

The main problem in face recognition is its high interclass variability. On one hand, it suffers from extrinsic variability, for instance the mapping from 2D to 3D or changes on illumination conditions cause that different views provide highly different realizations of the same face. On the other, intrinsic variability due to non-permanent face parameters, as skin color or facial hair length, adds information that is not useful into the recognition process. Thus, the key issue in face recognition is to extract only the meaningful features that characterize a human face, discarding all irrelevant attributes.

Generally speaking, a face recognition system for verification can be divided in the following stages:

1) Preprocessing

- *Localization and segmentation*
- *Normalization*

2) Face verification

- *Feature extraction*
- *Classification*

In the following sections, the implementation details for our frontal-view face recognition system are explained.

1) Preprocessing:

a) *Face location and segmentation*: Face detection and segmentation was performed by OpenCV face detector [3]. Based on cascade Haar classifiers, it provides excellent results in our scenario: a single user in front of a camera. It returns a bounding box centered on the detected face (see Figure 1).

b) *Normalization*: On the results presented on this paper only size normalization of the extracted faces was used. All face images were resized to 150x150 pixels, applying a bicubic interpolation if needed. After this stage, the image was cut on the borders (30 pixels on the upper and lower borders, and 10 into the left and the right ones), resulting into 90x130 pixel images, to discard most of the hair (a highly variant part of the face) and the picture background.

Although not integrated in the final system, we also developed a position correction algorithm based on detecting the eyes into the face and applying a rotation and resize to align the eyes of all pictures in the same coordinates.

The eye detection proposed in this work is based on a k-means clustering method in a bidimensional space [13]. Initially, the face is binarized and inverted, and the algorithm is not applied to the whole image but to an eye mask including only the upper half part. After that, the pixels are grouped into four clusters, using k-means method. Selecting the lower clusters of each side of the face the position of the eyes is estimated, as can be seen in Figure 3. Some results from different users are shown in Figure 4.



Fig. 1. Face extraction example from our database video sequences performed by OpenCV face detector. The gray scale size-normalized extracted face is shown on the upper left corner of the image.

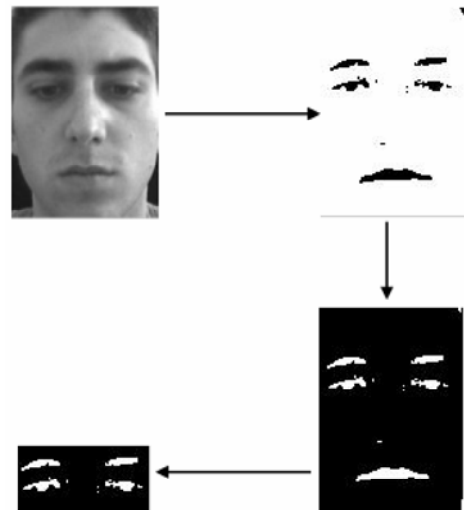


Fig. 2. Binarization, inversion and eye mask selection from detected and segmented face image.

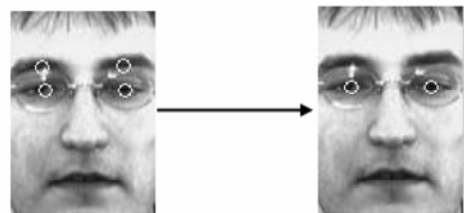


Fig. 3. Detecting and selecting clusters for eye detection.



Fig. 4. Eye detection examples for different users.



Fig. 5. Left: Mean of two different face images from the same user without position correction. Right: Mean of the same two images after position correction based on eye detection and location.

The orientation and size correction minimizes the diffusion in the eigenface conformation (see Feature Extraction section) and we believe that improves the verification rate. To illustrate the advantages of further normalization, the average of two images from the same user without and with position correction is shown in Figure 5.

Other normalization schemes would include removing luminance inhomogeneities. In our database, luminance conditions were approximately constant, hence no method was used for this purpose.

2) Face Verification:

a) Feature extraction: The features extracted were based on the Karhunen-Loeve (KL) expansion, also known as principal component analysis (PCA). The main reasons to use KL expansion was that it has been exhaustively studied and have proved to be quite invariant and robust when proper normalization is applied over the faces [1]. On the other hand, the main disadvantages of KL methods is its complexity and that the extracted base is data-dependent: if new images are added to the database the KL base need to be recomputed.

The main idea is to decompose a face picture as a weighted combination of the orthonormal base provided by the KL transform. The base corresponds to the eigenvectors of the covariance matrix of the data, known as eigenfaces (see Figures 6 and 7). This expansion is optimal in a MSE sense, meaning that the image reconstruction that minimizes the MSE, on a dimensional reduced space, is obtained removing



Fig. 6. Upper left corner: mean face image from the whole face database. From left to right, the whole database eigenfaces associated with the 7th largest eigenvalues are shown in decreasing order.

the eigenfaces associated with the smallest eigenvalues of the covariance matrix.

Thus, the decomposition of a face image into an eigenface space provides a set of features. The maximum number of features is restricted to the number of images used to compute the KL transform, although usually only the more relevant features are selected, removing the ones associated with the smallest eigenvalues. Two different approaches, database common eigenfaces and independent user eigenface space are detailed in the next sections.

Common Eigenface Space

In the classic eigenface method, proposed by Turk and Pentland [14], the PCA is performed on a dataset of face images from all users to be recognized.

The first step is to vectorize the set of N face images from different users in the database, F_1, \dots, F_N , resulting into a new set of vectors f_1, \dots, f_N . They can be written as a matrix, concatenating all images as columns,

$$X = [f_1, \dots, f_N] \quad (1)$$

Hence, removing the mean of the training vectors, f_μ , the data covariance matrix, $X^T X$, can be computed. Grouping as columns the k eigenvectors associated with the first largest eigenvalues into the matrix U , a k -dimensional feature vector for each image can be obtained as

$$y = U^T (f - f_\mu) \quad (2)$$

The feature vector y describes the contribution of each eigenface in representing the input face. Consequently, an image can be projected into the common eigenface space, generating a k -dimensional point.

User Eigenface Space

This approach is based on the same principles as standard PCA, explained in the previous section. The difference is that an eigenface space is extracted for each user. Thus, when a



Fig. 7. Two different examples of individual user eigenfaces. In each row, the first 4 eigenfaces for the same user are shown, the first one including the mean face of the user.

claimant wants to verify its identity, its vectorized face image is projected exclusively into the claimed user eigenface space and the corresponding likelihood is computed.

The advantage of this new approach is that it allows a more accurate model of the user's most relevant information, where the first eigenfaces are directly the most representative user's face information.

Another interesting point of this method is its scalability in terms of the number of users. Adding a new user or new pictures of an already registered user only requires to compute or recompute the specific eigenface space, but not the whole dataset base as in the standard approach. For verification systems, the computation of the claimant's likelihood to be an specific user is independent on the number of users in the dataset. On the contrary, for identification systems, the number of operations increases in a proportional way with the number of users, because as many projections as different users are required.

In the verification system described in this article, the independent user eigenface approach has been chosen. Each user's eigenface space was computed which 200 non-consecutive frames extracted from the described database videos.

b) Classification: For classification purposes, a GMM based classifier was used [12]. A total number of 10 non-consecutive images, not previously included into the training database, were used in each claim to compute the average log-likelihood of the claimant being the claimed user. Further details in GMM models and log-likelihood can be found in Section III-A.

C. Signature

Following Plamondon and Lorette [5], the methods of handwriting processing can be classified regarding the type of data acquisition – off-line vs. on-line. For off-line processing, the data acquisition is carried out from the Writing surface (e.g. paper) after the writing process. In the normal case, this off-line acquisition is done with an optical scanner device and the resulting data are a kind of 2-dimensional image. In contrast, in the on-line approach, the data acquisition occurs during

the writing process itself. The resulting data of this approach are signals, which describe the pen motion on the writing surface. For gathering of on-line handwriting data, special devices are used, for example graphic digitizer tablets, Tablet PCs or PDA-like computers with pressure sensitive screens. In the following we will concentrate on on-line handwriting processing, recorded by digitizer tablet devices.

The device we used for data acquisition is able to output the pen tip position on the active writing surface with a high resolution. Additionally it measures the pen pressure. The sampling rate is about 100Hz. (For details, see section IV.)

The raw sample point, captured at time t_i , is the following: $s_i = (x_t, y_t, p_t, t_i)$, where x_i , y_i and p_i are the pen tip position and the pressure, respectively. Figure 8 shows x-, y- and p-signals of an example signature.

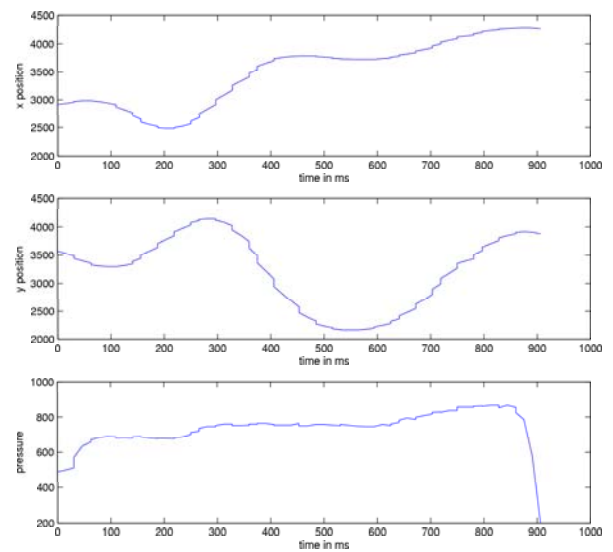


Fig. 8. Signals (x-, y-position and pen pressure) of one signature fragment.

In addition to these raw data, the writing velocity v_i as well as the tangent angle θ_i at every time t_i is computed:

$$v_i = \sqrt{\dot{x}_i^2 + \dot{y}_i^2} \quad \theta_i = \arctan(\dot{y}_i, \dot{x}_i)$$

with $\dot{x}_i = x_i - x_{i-1}$ and $\dot{y}_i = y_i - y_{i-1}$ (see [9]). These five dimensional feature vectors are used for GMM processing (see section III-A).

III. FUSION

It is well documented that multiple modalities are necessary for high performance in user verification and identification systems [2] [10]. As a consequence of this, a generic biometric system has four substantial modules

- sensor module* where raw biometric data are captured from the devices;
- feature module* in which a feature set is extracted from the raw data of each modality;
- matching module* where a classifier is utilized to compare the features extracted from the previous module with the trained patterns;

- (d) *decision module* in which the outputs of the classifiers are combined in order to make a decision.

In the following subsections the matching and decision modules are discussed.

Because of the use of multiple modalities, fusion techniques should be established for coupling the different modalities. Integration of information in a Multimodal biometric system can occur in different levels

- (a) *feature level* where the feature sets of different modalities are combined. Fusion at this level provides the highest flexibility but classification problems may arise due to the large dimension of the combined(concatenated) feature vectors.
- (b) *score (matching) level* is the most common level where the fusion take place. The scores of the classifiers are usually normalized and then they are combined in a consistent manner.
- (c) *decision level* where the output of the classifiers establish the decision via techniques such as majority voting. Fusion at the decision level is considered to be rigid for information integration.

The fusion of our system is done at the score level.

A. Matching Module

In the feature module, a feature set is extracted from each modality. The feature vectors of each modality constitute an D-dimensional feature space. Feature vectors with class labels—in our case one user constitute one class— can be used to estimate a model describing a class.

We propose a method similar to Bayesian classification for the determination of users' identification(or verification). The scores of each modality will be the posterior probabilities or decision risks calculated from the probabilities of the model. The posterior probability of pattern \mathbf{x} to belong in class ω_k can be computed with the Bayes rule

$$P(\omega_k|\mathbf{x}) = \frac{p(\mathbf{x}|\omega_k)P(\omega_k)}{p(\mathbf{x})}$$

where $p(\mathbf{x}|\omega_k)$ is the probability density function of class ω_k , $P(\omega_k)$ is the prior probability and $p(\mathbf{x})$ is merely a scaling signatures factor. The major problem in Bayesian classifier is the determination of $p(\mathbf{x}|\omega_k)$. Some assumptions have to be made about the structure of the class-conditional probabilities $p(\mathbf{x}|\omega_k)$.

One very common approach for approximating the unknown class-conditional probabilities $p(\mathbf{x}|\omega_k)$ is by using Gaussian Mixture Models(GMMs). A GMM is defined as

$$p(\mathbf{x}|\omega_k; \Theta) = \sum_{k=1}^C \alpha_k \mathcal{N}(\mathbf{x}; \mu_k, \Sigma_k)$$

where $\mathcal{N}(\mathbf{x}; \mu_k, \Sigma_k)$ is the Gaussian probability function with mean value μ_k and covariance matrix Σ_k , α_k are positive weights of the component k and $\sum_{k=1}^C \alpha_k = 1$. The parameter list

$$\Theta = \{\alpha_1, \mu_1, \Sigma_1, \dots, \alpha_C, \mu_C, \Sigma_C\}$$

defines a particular Gaussian mixture probability density function.

The parameters of the Gaussian mixture probability density functions are estimated with Expectation Maximization(EM) algorithm [4]. EM algorithm is an iterative method for calculating maximum likelihood distribution parameters. It can also be used to handle cases where an analytical approach for maximum likelihoods estimation is infeasible, such as GMMs with unknown and unrestricted covariance matrices and means.

The training vectors used in EM are first normalized making the standard deviation of each class equal to unity. Given a pattern $\mathbf{x} = [x_1 \ x_2 \ \dots \ x_D]$ the new pattern is defined as

$$\mathbf{x}' = [x'_1 \ x'_2 \ \dots \ x'_D] = [x_1/\sigma_1 \ x_2/\sigma_2 \ \dots \ x_D/\sigma_D]$$

Moreover a Universal Background Model (UBM) [7] is applied in order to model the user-independent distribution of the features. The class-conditional probabilities(or likelihoods) computed by the UBM used for the normalization of the users' class-conditional probabilities. The normalization is done by dividing the likelihood of the UBM from the likelihood of the user.

B. Decision Module

Several techniques have been used to consolidate the matching scores and arrive at a decision. There are two major categories

- (a) *classification techniques* where a feature vector is constructed using the matching scores output by the individual classifier. Typical examples are Neural Networks, Decision Trees and Support Vector Machines;
- (b) *combination techniques* where the output of the classifiers are combined accordingly. Simple yet considerable examples are Sum or Product Rules and Linear combination of the scores.

In this work we concentrate on the combination techniques.

The advantage of using GMMs for obtaining the matching scores for all the modalities is that they are homogeneous. Applying Bayes rule all the scores are the class-conditional probabilities of the models. If we assume that the a priori probabilities are equiprobable for each user and apply a normalization scheme then the scores are the posterior probabilities. To obtain the posterior probabilities it is sufficient to divide the likelihood of each model with the sum of the likelihoods of all the models. In mathematical terms,

$$P(\omega_k|\mathbf{x}) = \frac{p(\mathbf{x}|\omega_k)}{\sum_{i=1}^C p(\mathbf{x}|\omega_i)} \quad k = 1, \dots, C$$

This operation actually makes the likelihoods $p(\mathbf{x}|\omega_k)$ a distribution ,i.e. likelihoods are transformed in posterior probabilities $P(\omega_k|\mathbf{x})$.

After normalizing the scores of each modality with the above method a simple product rule is applied. This rule is based on the assumption of independence of the modalities. In general, different biometric traits of an individual are mutually independent. There are also other normalization methods as well as combination techniques that were tested but they did not perform better.

IV. DATABASE

A. Database of signatures

The device used for recording the handwriting data was a Wacom Graphire3 digitizing tablet. Size of sensing surface is 127.6mm x 92.8mm. With spatial resolution of 2032 lpi (lines per inch), able to measure 512 degrees of pressure. Data is acquired with a non-fixed sampling rate of about 100Hz.

Altogether, the new database consists of 1641 signatures of 47 persons. For each person, at least 30 signatures are available. The structure of the database is as follows:

```
signatures+-user01+-2005-08-08-12-00-00.dat
|          +-2005-08-08-12-01-00.dat
|          +- ...
|
+-user02+-2005-08-08-13-01-00.dat
|          +-2005-08-08-13-04-00.dat
|          +- ...
|
+-user03+- ...
|          +- ...
|
+-user47+- ...
```

Each .dat file represents one signature. Each line of a .dat file consists of four comma separated integer values for the sampled x- and y-position of the pen tip, the pen pressure and the timestamp (in ms). Those lines with values of -1 for x, y and pressure represent a pen-up/pen-down event.

Because of hypothetical legal and privacy concerns, the definitive acquired handwritten inputs were not real signatures. At an initial stage, experiments were done with a preliminary database composed of real signatures from our team members; then, test subjects were asked to write an arbitrary word as *fake signature*, other subjects chose to do a modification of their true signature. Every subject had to repeat the writing at least thirty times. They were able to see their writings on the screen.

Forged signatures

As a test for the robustness of the identity verification system, *skilled forgeries* of the real signatures of the preliminary signature database were created. The choice of the considered modalities for this database was done admitting that speech and face are very hard to reproduce, in comparison to signatures. For time constraints, *skilled forgeries* were not added to the definitive database of *fake signatures*.

For helping the imitators to reproduce the signatures, an application was developed. It consists of a user interface written in Matlab. Its first task is to reproduce the image of the signature the user wants to be imitate. Then, the user can play a movie representing the exact way the signature has been drawn, in function of time. The speed of the signature is so conserved; and the user can modify it as a parameter for playing the movie. The second parameter of this application is the number of frames per second. Once these parameters are set by the user, the movie is created with linear interpolation between successive samples.

B. Database of audio and video

Audio and still pictures are extracted from the video, which is encoded in raw UYVY. AVI 640 x 480, 15.00 fps with uncompressed 16bit PCM audio; mono, 32000 Hz little endian. A few videos are with uncompressed PCM audio; stereo, 44100 Hz little endian.

We provide Perl scripts for extraction of audio and still pictures from the videos, extraction of audio takes significantly less time than picture extraction. These scripts use `mplayer` and `sox`.

Uncompressed PNG files are extracted from the video files for feeding the face detection algorithms.

Audio is extracted as 16 bit PCM WAV file (with wav header), sampled at 16000 Hz, mono little endian.

Capturing Devices

- Allied Vision Technologies AVT marlin MF-046C 10 bit ADC, 1/2" (8mm) Progressive scan SONY IT CCD.
- Shure SM58 microphone. Frequency response 50 Hz to 15000 Hz. Unidirectional (Cardioid) dynamic vocal microphone.

Bugs

There was a problem with `mplayer` not writing the `byte_alignment` of the wav header correctly, which caused files not being read correctly on Matlab. A patch fixing the bug was sent and merged in the `mplayer` CVS. We include the patch with the database. Against CVS revision: 1.29 of `/cvsroot/mplayer/main/libao2/ao_pcm.c` The patch should also work against `MPlayer-1.0pre7`, which was the latest official release of `mplayer` which was unpatched. So using the CVS version is recommended, until a patched official release is made. If not, `--fixheader` option can be used for separate audio from video stream* scripts.

V. RESULTS

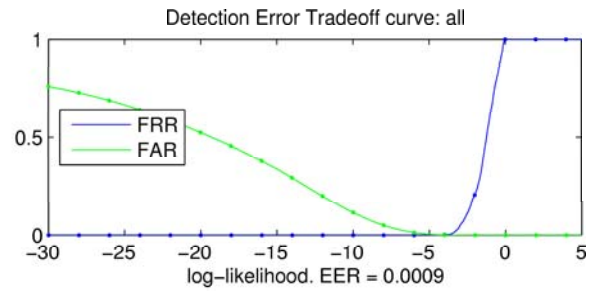
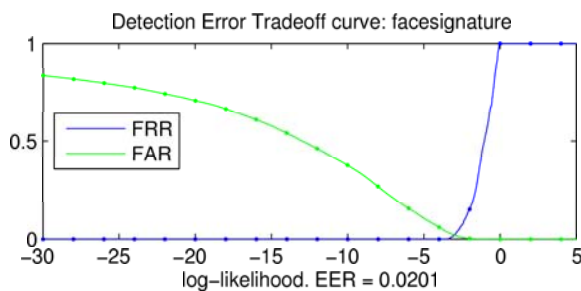
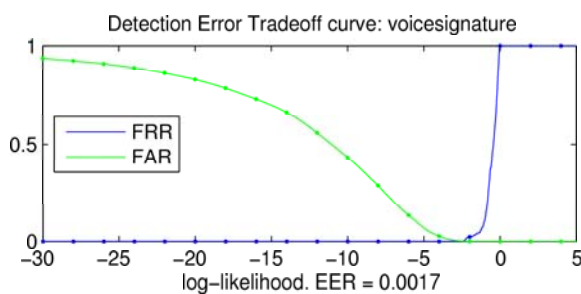
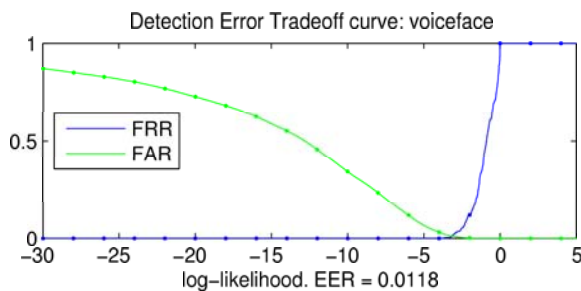
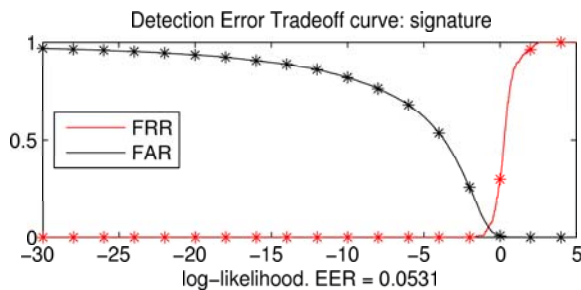
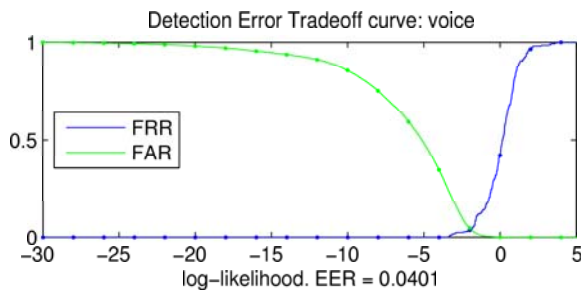
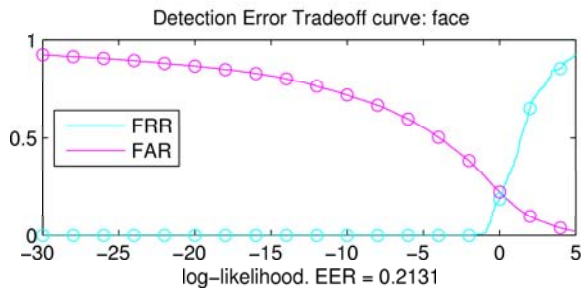
In a verification system we often have a tradeoff between the ratio of impostors accepted on the system, which is denoted as False Acceptance Rate (FAR) or false positives, and the ratio of rejected genuines, denoted as False Rejection Rate (FRR) or false negatives. When choosing the decision threshold θ , choosing a too low value, would let too many impostors in, while choosing a very high value would cause too many rejections to genuines, and the performance of the system would be unacceptable for practical uses.

A good decision is to choose θ_{EER} such as $FAR(\theta) = FRR(\theta)$, but since FAR and FRR are discrete, an option is to choose:

$$\theta_{EER} = \operatorname{argmin}_{\theta} |FRR(\theta) - FAR(\theta)|$$

$$EER(\theta) = \frac{FRR(\theta) + FAR(\theta)}{2}$$

To compare the performance of our system we use the Detection Error Tradeoff (DET) curve and the definitions above for the calculation of the Equal error rate (EER):



VI. CONCLUSION AND FUTURE WORK

We can see how combining all the modalities allows us to achieve an EER of 0.09% which is much better than those of the modalities taken separately.

Different methods of fusion could be tested for cases in which modalities could not be considered independent. Also new feature extraction methods could be tested.

ACKNOWLEDGMENT

The authors would like to thank eNTERFACE and its sponsors for providing the means which made this project possible. Also thanks to previous research articles and the free/open source software we have used, which has allowed us to *stand on the shoulders of giants*.

REFERENCES

- [1] Chellappa R., Wilson C.L., Sirohey S., Human and Machine Recognition of Faces: A Survey. Proceedings of the IEEE. Volume 83. Number 5. May 1995.
- [2] A.K. Jain, A. Ross, S. Prabhakar, *An introduction to biometric recognition*, IEEE Trans. Circuits Systems Video Technology, pp. 4–20, 2004.
- [3] Open Source Computer Vision Library Documentation. <http://www.intel.com/technology/computing/opencv/>
- [4] Pekka Paalanen, *Bayesian classification using gaussian mixcut model and EM estimation: implementation and comparisons*, Information Technology Project, 2004, <http://www.it.lut.fi/project/gmmbytes/>
- [5] R. Plamondon, G. Lorette, *Automatic Signature Verification and Writer Identification – The State of the Art*, Pattern Recognition, Vol. 22, No. 2, pp. 107–131, 1989.
- [6] D. Reynolds *Speaker identification and verification using Gaussian Mixture Models*, Speech Communication, 1995.
- [7] D.A. Reynolds, T.F. Quatieri and R.B. Dunn, *Speaker verification using adapted gaussian mixture models*, Digital Signal Processing, pp. 19–41, 2000.
- [8] D. Reynolds, T. Quatieri and R. Dunn, *Speaker Verification Using Adapted Gaussian Mixture Models*, Digital Signal Processing 19-41, 2000.
- [9] J. Richiardi and A. Drygajlo, *Gaussian Mixture Models for Online Signature Verification*, ACM Press, 2003.
- [10] A. Ross, A.K. Jain, *Information Fusion in Biometrics*, Pattern Recognition Letters, 2003.
- [11] A. Ross, and K. Jain, *Multimodal Biometrics: An Overview*, Prom. of 12th European Signal Processing Conf (EUSIPCO), pp. 1221-1224, 2004.
- [12] Sanderson C., Bengio S., Robust Features for Frontal Face Authentication in Difficult Image Conditions. IDIAP-RR 03-05, January 2003.
- [13] Seber, G. A. F., *Multivariate Observations*, Wiley, 1984.
- [14] Turk M., Pentland A., *Eigenfaces for Recognition*. Journal of Cognitive Neuroscience. Volume 3, Number 1. Massachusetts Institute of Technology, 1991.

Yannis Stylianou Associate professor in the Computer Science Department, University of Crete. email: yannis@csd.uoc.gr

Yannis Pantazis Postgraduate student in the Computer Science Department, University of Crete. email: pantazis@csd.uoc.gr

Felipe Calderero Technical University of Catalonia (UPC). Superior Telecommunication Engineering. email: felipe@gps.tsc.upc.edu

Pedro Larroy Technical University of Catalonia (UPC). Superior Telecommunication Engineering. email: pedro@larroy.com

Francois Severin Faculte Polytechnique de Mons. email: francois.severin@tcts.fpms.ac.be

Sascha Schimke Otto-von-Guericke University Magdeburg, Germany. email: sschimke@iti.cs.uni-magdeburg.de

Rolando Bonal Universidad de las Ciencias Informaticas (UCI), Habana, Cuba. email: rolandobonal@gmail.com

Federico Matta Institut Eurecom (CNRS) France. email: Federico.Matta@eurecom.fr

Athanasios Valsamakis Postgraduate student in the Computer Science Department, University of Crete. email: valsamak@csd.uoc.gr

The Speech Conductor: Gestural Control of Speech Synthesis

Christophe d'Alessandro (1), Nicolas D'Alessandro (2), Sylvain Le Beux (1), Juraj Simko (3),

Feride Çetin(4), Hannes Pirker (5)

(1) LIMSI-CNRS, Orsay, France, (2) FPMS, Mons, Belgium (3) UCD, Dublin, Ireland,

(4) Koç Univ., Istanbul, Turkey, (5), OFAI, Vienna, Austria

Abstract

The Speech Conductor project aimed at developing a gesture interface for driving (“conducting”) a speech synthesis system. Four real-time gesture controlled synthesis systems have been developed. For the first two systems, the efforts focused on high quality voice source synthesis. These “Baby Synthesizers” are based on formant synthesis and they include refined voice source components. One of them is based on an augmented LF model (including an aperiodic component), the other one is based on a Causal/Anticausal Linear Model of the voice source (CALM) also augmented with an aperiodic component. The two other systems are able to utter unrestricted speech. They are based on the MaxMBROLA and MidiMBROLA applications. All these systems are controlled by various gesture devices. Informal testing and public demonstrations showed that very natural and expressive synthetic voices can be produced in real time by some combination of input devices/synthesis system

Index Terms—speech synthesis, glottal flow, gesture control, expressive speech.

I. PRESENTATION OF THE PROJECT

A. Introduction

Speech synthesis quality seems nowadays acceptable for applications like text reading or information playback.

However, these reading machines lack expressivity. This is not only a matter of corpus size, computer memory or computer speed. A speech synthesizer using several times more resources than currently available will probably improve on some points (less discontinuities, more smoothness, better sound) but expression is made of real time subtle variations according to the context and to the situation. In daily life, vocal expressions of strong emotions like anger, fear or despair are rather the exception than the rule. Then a synthesis system should be able to deal with subtle continuous expressive variations rather than clear cut emotions. Fundamental questions concerning expression in speech are still unanswered, and to some point even not stated. Expressive speech synthesis is the next challenge. Expressive speech synthesis may be viewed from two sides: on the one hand is the question of expression specification (what is the suited expression in a particular situation?) and on the other hand is the question of expression realisation (how is the

specified expression actually implemented). The first problem (situation analysis and expression specification) is one of the most difficult problems for research in computational linguistics because it involves deep understanding of the text and its context. Without a deep knowledge of the situation defining an adequate expression is difficult, if not impossible.

It is only the second problem that has been addressed in this workshop. Given the expressive specifications, produced and controlled in real time by a “speech conductor”, given the intended expression, or an “expression score” for a given speech utterance, how to “interpret” the speech produced according to this intended expression?

The Speech Conductor project aims at developing and testing gesture interfaces for driving (“conducting”) a speech or voice synthesis system. The goal is to modify speech synthesis in real time according to the gestures of the “Speech Conductor”. The Speech Conductor adds expressivity to the speech flow using speech signal synthesis and modification algorithms and gesture interpretation algorithms. This multimodal project involves sounds, gestures and text.

B. Domains and challenges

The main goal of this project was to test various gesture interfaces for driving a speech synthesiser and then to study whether more “natural” expressive speech (as compared to rule-based or corpus-based approaches) could be produced. The problems addressed during the workshop were:

1. Identify the parameters of expressive speech and their relative importance. All the speech parameters are supposed to vary in expressive speech. In time domain a list of speech parameters would encompass: articulation parameters (speed of articulation, formant trajectories, articulation loci, noise bursts, etc.) phonation parameters (fundamental frequency, durations, amplitude of voicing, glottal source parameters, degree of voicing and source noise etc.). Alternatively, physical parameters (sub glottal pressure, larynx tension) or spectral domain parameters could be used.
2. Signal processing for expressive speech. Techniques for parametric modification of speech: fundamental frequency, duration, articulation, voice source.
3. Domain of variation and typical patterns for expressive speech parameters, analysis of expressive speech.

4. Gesture capturing and sensors. Many types of sensor and gesture interfaces were available. The most appropriate have been selected and tried.
5. Mapping between gestures and speech parameters. The correspondence between gestures and parametric modifications is of paramount importance. This correspondence can be more or less complex (one to many, many to one, one to one). A physiologically inspired model for intonation synthesis has been used.
6. Different types of vocal synthesis have been used. Parametric source/filter synthesis proved useful for accurately controlling voice source parameters. Diphone based concatenative speech synthesis proved useful for more unrestricted speech synthesis applications, but allowed for less fine grained controls. Of course real time implementations of the synthesis systems were needed.
7. Expression, emotion, attitude, phonostylistics. Questions and hypotheses in the domain of emotion research and phonostylistics, evaluation methodology for expressive speech synthesis have only marginally been addressed because of the short time available. For the same reason preliminary evaluation of the results obtained took place on an informal basis only.

C. Gesture Control Devices

Several devices, whose controllers and ranges are quite different, were used. At first, we used two keyboards, one Roland PC-200, with 49 keys, a Pitch Bend /Modulation Wheel and one fader. The range of the keyboard is by default between 36 and 84 but can be shifted in order to change the frequency register. The Pitch Bend/Modulation wheel sends values between 0 and 127 according to the MIDI protocol. Thus, these several controllers are respectively sending values on dedicated Note On/Off, Pitch Bend and Control Change channels.

The second keyboard was a Edirol PCR-50 which features 8 knobs and 8 faders in addition to the controls mentioned before. Similarly, in this keyboard the values are set between 0 and 127 and it sends data on several Control Change channels.

In addition to the Roland keyboard we also used an Eobody controller to have some extra knob controls in order to drive the MaxMBROLA Text-To-Speech synthesizer. This sensor interface converts any sensor raw data to MIDI protocol, but as a matter of fact we only used the inbox knobs. We were also able to use a MIDI foot controller providing ten switches in ten different banks and two expression pedals.

A P5 Glove with five flexion sensors linked to the fingers that could bend when fist clench was also employed. The sensors send data in range 0 to 63. Thanks to an Infrared sensor, the glove offers the ability to track the hand position in three spatial dimensions (x,y,z) within a continuous range roughly equal to [-500,+500].

The glove does not actually use MIDI protocol but Open Sound Control (OSC) instead. Contrary to MIDI which sets

data in a serial way, under OSC the values are sent in parallel, allowing a fixed rate for every controller.

D. Overview of the work done

The work has been organized along two main lines: text-to-speech synthesis and parametric voice quality synthesis. As for text-to-speech synthesis two different configurations have been produced. For one of the systems the only parameter controlled in real time is fundamental frequency. Phonemes and durations are computed automatically by the text-to-speech engine (we used Mary (Schröder & Trouvain, 2003) for English) and then produced by the MBROLA diphone system (Dutoit & al., 1996). For the second system, syllables are triggered by the player. Then durations, fundamental frequency and intensity are controlled using the MidiMBROLA synthesis system (D'Alessandro & al. 2005).

As for parametric voice quality synthesis, coined herein the "Baby Synthesizers" also two different approaches have also been implemented. Both are based on a parametric description of the voice source. In one system, the well-known LF model (Fant & al. 1985, Fant 1995) of the glottal flow derivative has been used, and augmented with an aperiodic component. The other system is based on a spectral approach to glottal flow modelling, the Causal/Anticausal Linear Model, CALM (Doval & al. 2003). This model has also been augmented with an aperiodic component.

In the remaining of this paper, the four systems developed during the workshop will be described in more detail.

II. REAL TIME CONTROL OF AN AUGMENTED LF-MODEL.

A. The voice source model in the time domain

In the linear acoustic model of speech production, the effect of the voice source is represented by the time-varying acoustic flow passing through the glottis. When the vocal folds are regularly oscillating (voiced speech), the glottal flow can be represented using a glottal flow model, the most widely used being the Liljencrants-Fant (LF) model (Fant & al. 1985). The glottal flow is the air stream coming from the lungs through the trachea and pulsed by the glottal vibration. All the glottal flow models are pulse like, positive (except in the case of ingressive speech), quasi-periodic, continuous, and differentiable (except at closure). Acoustic radiation of speech at the mouth opening can be approximated as a derivation of the glottal flow. Therefore, the glottal flow derivative is often considered in place of the glottal flow itself. The form of the glottal flow derivative can often be recognized in the speech waveform, with additional formant ripples. The time-domain glottal flow models can be described by equivalent sets of 5 parameters (Doval & d'Alessandro, 1999):

- A_v : peak amplitude of the glottal flow, or amplitude of voicing.
- T_0 : fundamental period (inverse of F_0)
- O_q : open quotient, defined as the ratio between the glottal open time and the fundamental period. This

quotient is also defining the glottal closure instant at time $O_q * T_0$.

- A_m : asymmetry coefficient, defined as the ratio between the flow opening time and the open time. This quotient is also defining the instant T_m of maximum of the glottal flow, relative to T_0 and O_q ($T_m = A_m * O_q * T_0$). Another equivalent parameter is the speed quotient S_q , defined as the ratio between opening and closing times, $A_m = S_q / (1 + S_q)$.
- Q_a : the return phase quotient defined as the ratio between the effective return phase duration (i.e. the duration between the glottal closure instant, and effective closure) and the closed phase duration. In case of abrupt closure $Q_a = 0$.

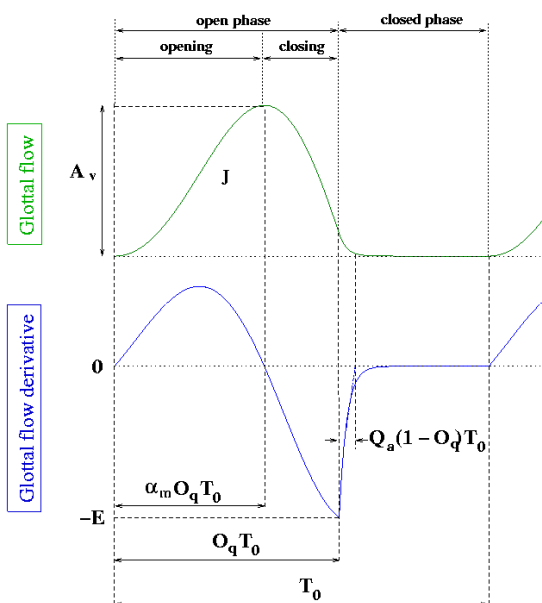


Figure 1: Time domain models of the glottal flow and glottal flow derivative (LF-model), after Henrich & al. 2002.

When considering the glottal flow derivative, the peak amplitude is generally negative, because the closing phase is generally shorter than the opening phase. So the descending slope of the glottal flow is steeper, and its derivative larger. All the time-domain parameters are equivalent for the glottal flow and its derivative except this amplitude parameter:

- E : peak amplitude of the derivative, or maximum closure speed of the glottal flow. Note that E is situated at $O_q * T_0$, or glottal closure instant. It is often assumed that E represents the maximum acoustic excitation of the vocal tract

E and A_v are both representing a time domain amplitude parameter. One or the other can be used for controlling amplitude, but E appears more consistently related to loudness and should probably be preferred for synthesis. The waveform and derivative waveform of the LF model are plotted in Figure

1. It must be pointed out that an aperiodic component must also be added to the periodic LF model. Two types of aperiodicities have to be considered: structural aperiodicities (jitter and shimmer) that are perturbations of the waveform periodicity and amplitude, and additive noise.

Note that compared to the LF model new parameters are added for controlling the aperiodic component. Shimmer and Jitter are perturbation of T_0 amplitude of the LF model (structural aperiodicities). Filtered white noise is also added to the source for simulating aspiration noise in the voice source. The voice source waveform is then passed in a vocal tract filter to produce vowels. The initial formant transitions have been designed to produce a voiced stop consonant close to /d/. This time-domain “baby synthesizer” based on the augmented LF model is presented in Figure 2. The circles indicate those parameters that can be controlled in real time by the gesture captors

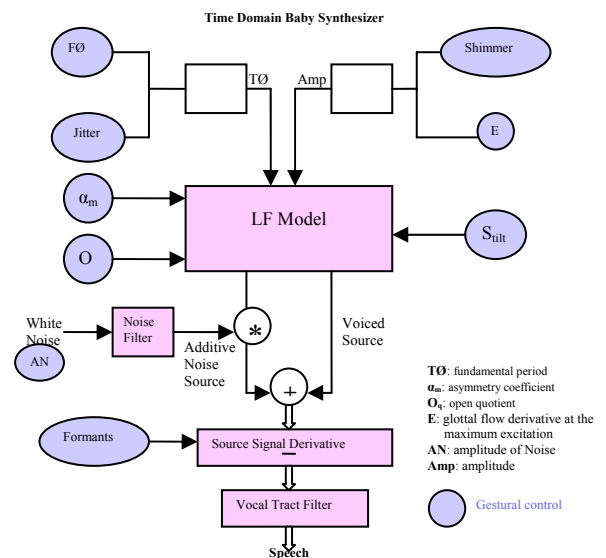


Figure 2. The time-domain “baby synthesizer” implemented in the project, LF model of the source, source aperiodicities and vocal tract filter.

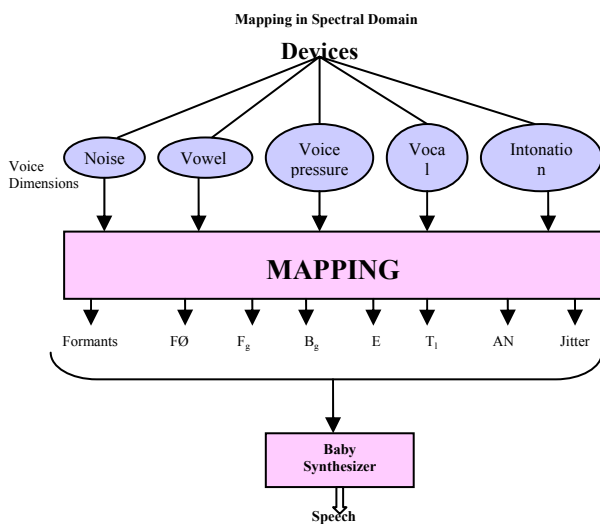
B. Mapping

There is no one-to-one correspondence between voice quality and glottal flow parameters. Their relationships are the subject of a large body of work. They can be sketched as follows (d’Alessandro, forthcoming). F_0 describes melody. A very low F_0 generally signals creaky voice and a high F_0 generally signals falsetto voice. O_q describes mainly the lax-tense dimension. O_q is close to 1 for a lax voice, and may be as low as 0.3 for very pressed or tense phonation. As A_v represents the maximum flow, it is an indication of flow voice, and it may help for analysis of the vocal effort dimension. E correlates well with the sound intensity. Q_a

correlates also with the effort dimension. When $Q_a = 0$ the vocal cords close abruptly. Then both E the asymmetry A_m are generally high, and so is vocal effort. Conversely, large values of Q_a (0.05-0.2) give birth to a smooth glottal closure –the vocal effort is low. The asymmetry coefficient A_m has an effect on both the lax-tense dimension (asymmetry is close to 0.5 for a lax voice, and higher for a tense voice) and the vocal effort dimension (asymmetry generally increases when the vocal effort increases). Therefore some sort of mapping between raw voice source parameters and voice quality dimensions is needed.

For controlling of the baby synthesizers, voice quality dimensions are mapped onto voice source acoustic parameters. These voice quality dimensions are then controlled by the gesture captors, as explained in Figure 3.

Figure 3. Mapping in Time domain



C. Gestural control

The augmented LF model has been implemented entirely in the Pure Data environment. The implementation is based on the normalized LF model worked out in (Doval & d'Alessandro 1999).

The way controllers have been mapped to the various synthesizers was somewhat arbitrary. It must be pointed out that controllers could practically be driving any of the several synthesizers we implemented. For the augmented LF model Baby synthesizer the configuration was settled as follows:

- The Edirol MIDI keyboard was driving three voice dimensions. The keys from (from left to right) define the vocal effort, and the velocity of the pressed key was linked to the glottal pressure.
- In order to be able to have a dynamic mapping of these two dimensions we chose to have the possibility to change the parameters driving these dimensions. So that we could easily set the mid value and the span of asymmetry, open quotient and closing phase time, these parameters were each set by two knobs.

- The Pitch Bend/Modulation wheel was respectively controlling Frequency and Volume in such a way that no sound is produced the wheel is released.
- In addition to this, we used the pedal board to switch between the different presets of the vocal tract formants of different predefined vowels (a,e,i,o,u).
- Finally, one expression pedal of this pedal board was used to add noise to the signal generated.

III. REAL TIME CONTROL OF A CAUSAL/ANTICAUSAL LINEAR SPECTRAL MODEL

A. The voice source model in the spectral domain

Modelling the voice source in the spectral domain is interesting and useful because the spectral description of sounds is closer to auditory perception. Time-domain and frequency domain descriptions of the glottal flow are equivalent only if both the amplitude and the phase spectrum are taken into account, as it is the case in this work.

The voice source in the spectral domain can be considered as a low-pass system. It means that the energy of the voice source is mainly concentrated in low frequencies (recall that only frequencies below 3.5 kHz were used in wired phones) and is rapidly decreasing when frequency increases. The spectral slope, or spectral tilt, in the radiated speech spectrum (which is strongly related to the source derivative) is at most -6 dB/octave for high frequencies. As this slope is of +6 dB/octave at frequency 0, the overall shape of the spectrum is a broad spectral peak. This peak has a maximum, mostly similar in shape to vocal tract resonance peaks (but different in nature). This peak shall be called here the “glottal formant”. This formant is often noticeable in speech spectrograms, where it is referred to as the “voice bar”, or glottal formant below the first vocal tract formant.

Spectral properties of the source can then be studied in terms of properties of this glottal formant. These properties are:

1. the position of the glottal formant (or “frequency”);
2. the width of the glottal formant (or “bandwidth”);
3. the high frequency slope of the glottal formant, or “spectral tilt”;
4. the height of the glottal formant, or “amplitude”.

One can show that the frequency of the glottal formant is inversely proportional to the open quotient O_q (Doval et al. 1997). It means that the glottal formant is low for a lax voice, with a high open quotient. Conversely, a tense voice has a high glottal formant, because open quotient is low.

The glottal formant amplitude is directly proportional to the amplitude of voicing. The width of the glottal formant is linked to the asymmetry of the glottal waveform. The relation is not simple, but one can assume that a symmetric waveform (a low S_q) results in a narrower and slightly lower glottal formant. Conversely, a higher asymmetry results in a broader and slightly higher glottal formant.

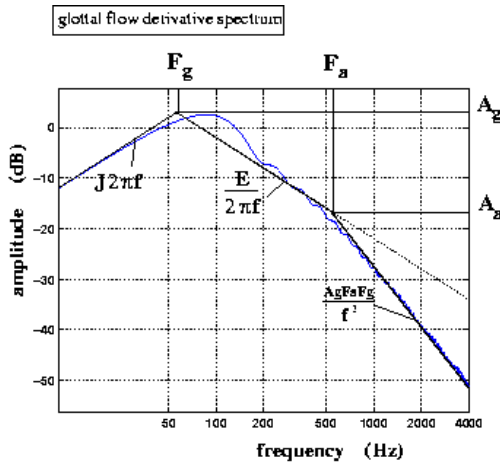


Figure 4. Glottal flow derivative spectrum (after Henrich & al. 2002)

Around a typical value of the asymmetry coefficient (2/3) and for normal values of open quotient (between 0.5 and 1), the glottal formant is located slightly below or close to the first harmonic ($H_1 = f_0$). For $O_q=0.4$ and $A_m=0.9$, for instance, it can then reach the fourth harmonic

Up to now, we have assumed an abrupt closure of the vocal folds. A smooth closure of the vocal folds is obtained by a positive Q_a in time domain. In spectral domain, the effect of a smooth closure is to increase spectral tilt. The frequency position where this additional attenuation starts is inversely proportional to Q_a . For a low Q_a , attenuation affects only high frequencies, because the corresponding point in the spectrum is high. For a high Q_a , this attenuation changes frequencies starting at a lower point in the spectrum.

In summary, the spectral envelope of glottal flow models can be considered as the gain of a low-pass filter. The spectral envelope of the derivative can then be considered as the gain of a band-pass filter. The source spectrum can be stylized by 3 linear segments with +6dB/octave, -6dB/octave and -12dB/octave (or sometimes -18dB/oct) slopes respectively. The two breakpoints in the spectrum correspond to the glottal spectral peak and the spectral tilt cut-off frequency. An example displaying linear stylization of the envelope of the glottal spectrum in a log representation is given in Figure 4.

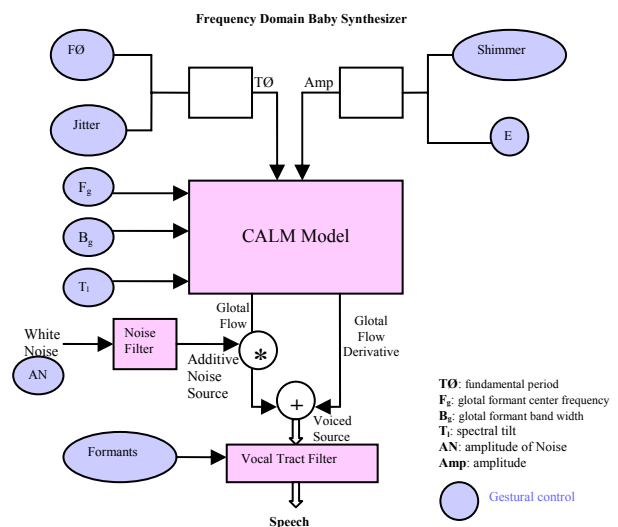
For synthesis in the spectral domain, it is possible to design an all-pole filter which is comparable to e.g. the LF model. This filter is a 3rd order low-pass filter, with a pair of conjugate complex poles, and a simple real pole. The simple real pole is given directly by the spectral tilt parameter. It is mainly effective in the medium and high frequencies of the spectrum. The pair of complex-conjugate poles is used for modeling the glottal formant. If one wants to preserve the glottal pulse shape, and then the glottal flow phase spectrum, it is necessary to design an anticausal filter for this poles pair. If one wants to preserve the finite duration property of the glottal pulse, it is necessary to truncate the impulse response of the filter. The spectral model is then a Causal (spectral tilt) Anti-causal (glottal formant) Linear filter Model (CALM, see Doval & al. 2003). This model is computed by filtering a pulse train by a

causal second order system, computed according to the frequency and bandwidth of the glottal formant, whose response is reversed in time to obtain an anti-causal response. Spectral tilt is introduced by filtering this anti-causal response by the spectral tilt component of the model. The waveform is then normalized in order to control accurately the intensity parameter E .

An aperiodic component is added to this model, including jitter, shimmer and additive filtered white noise. The additive noise is also modulated by the glottal waveform.

Then the voice source signal is passed through a vocal tract formant filter to produce various vowels. Figure 6 presents an overview of the spectral “Baby synthesizer”.

Figure 6. CALM Model



B. Mapping

This global spectral description of the source spectrum shows that the two main effects of the source are affecting the two sides of the frequency axis. The low-frequency effect of the source, related to the lax-tense dimension is often described in terms of the first harmonic amplitudes H_1 and H_2 or in terms of the low frequency spectral envelope. A pressed voice has a higher H_2 compared to H_1 , and conversely a lax voice has a higher H_1 compared to H_2 . The effort dimension is often described in terms of spectral tilt. A louder voice has a lower spectral tilt, and spectral tilt increases when loudness is lowering.

Then the vocal effort dimension is mainly mapped onto the spectral tilt and glottal formant bandwidth parameters (asymmetry), although the voice pressure dimension depends mostly on the glottal formant centre frequency, associated to open quotient.

Other parameters of interest are structural aperiodicities (jitter and shimmer) and additive noise.

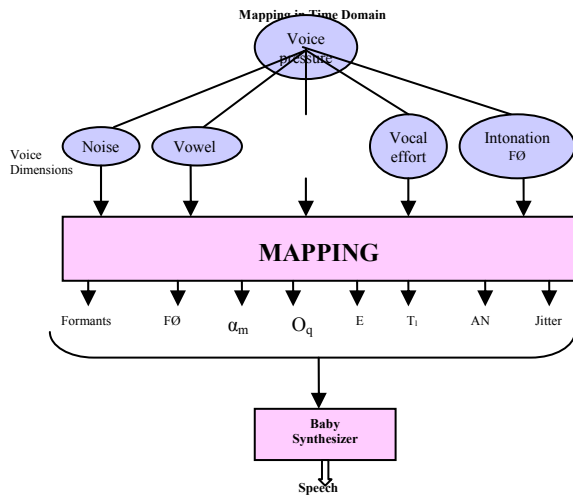


Figure 7. Mapping in Spectral domain

C. Gestural control of the spectral Baby Synthesizer

For this synthesizer, a P5 data glove is used. This input device allows driving 8 continuous variable parameters at once: 3 spatial position x , y , z associated with the movement of the glove relative to a fixed device on the table and 5 parameters associated with bending of the five fingers. Several keys on the computer keyboard are controlling vowels. The glove was driving the spectral-domain glottal source model. Only the two horizontal spatial dimensions (x, z) were used as follows: the x variable was linked to intensity E and the z variable was linked to fundamental frequency. All the fingers but the little finger were used to control respectively (beginning from the thumb) noise ratio, Open Quotient, Spectral Tilt and Asymmetry. This mapping is most reliable and effective (compared to the keyboard used in the first experiment). Only a short training phase was sufficient to obtain very natural voice source variations. The computer keyboard was used for changing values of the formant filters for synthesizing different vowels, and then basic vocal tract articulations.

IV. REAL TIME CONTROL OF F0 IN A TEXT-TO-SPEECH SYSTEM USING MAXMBROLA

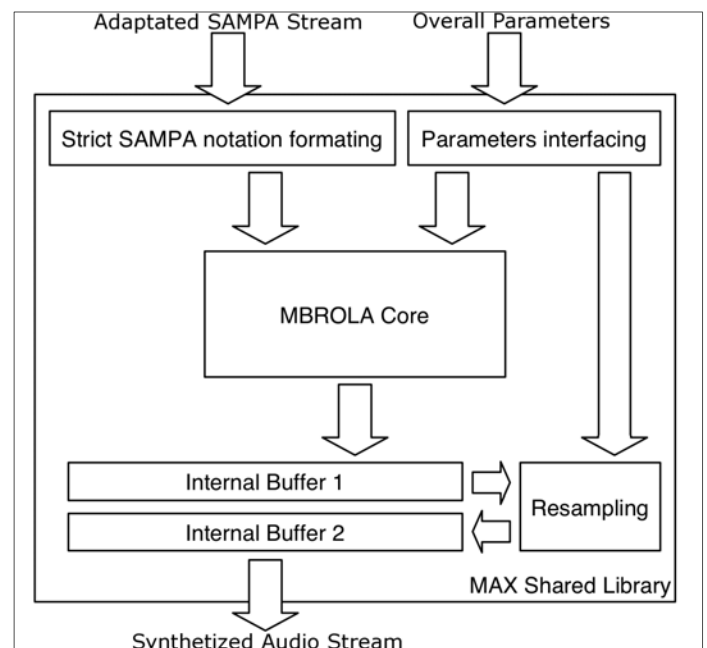
A. Max/MSP Graphical Programming Environment

The Max graphical development environment and its MSP audio processing library (Zicarelli & al., 2004) are widely used the computer music community. This software is a powerful tool in many fields of electronic music like real-time sound processing, control mapping, composition, enhancement of performance abilities etc. It is a rare example of an intuitive interface (design of personalized modules by the building of graphs of simple functions, called *objects*) and a high level of flexibility (functions accepting and modifying numbers, symbols, audio and video stream, etc) at the same time. The capabilities of that software increase every day due to the help of an active developer community providing new *external* objects (or *externals*).

B. MaxMBROLA~ external object: MBROLA inside Max/MSP

This section explains how the MBROLA technology has been integrated inside the Max/MSP environment (D'Alessandro & al. 2005). Max/MSP objects work as small servers. They are initialized when they are imported into the workspace. They contain a set of dedicated functions (methods) which are activated when the object receives particular messages. These messages can be simple numbers, symbols or complex messages with a header and arguments. Considering that real-time request-based protocol of communication between objects, a Max/MSP external object containing the MBROLA algorithm has been developed and a particular set of messages (header and arguments) has been formalized to communicate with the synthesizer.

Figure 8. Internal structure of the MaxMBROLA~ external object (after D'Alessandro & al. 2005).



As shown in Figure 8, we can separate the possible requests in two main channels. On one side, there is parameter modification, which influences the internal state of the synthesizer. On the other side, there is the phonetic/prosodic stream, which generates speech instantaneously.

C. Available actions of the object

1) Internal state modifications

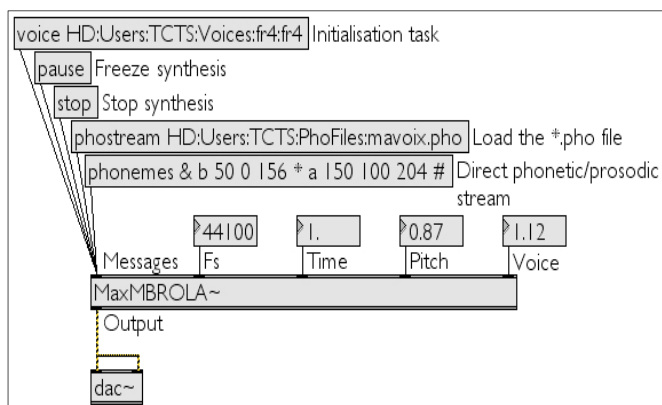
Specific modifications of the internal state of the MBROLA synthesizer can be applied with Max/MSP requests. Here follows a description of the supported actions. The labels are used to name inlets (from left to right: *Messages*, *Fs*, *Time*, *Pitch* and *Voice*) and examples of the supported messages are illustrated on Figure 9.

The synthesizer always starts with the initialization task (*Messages* inlet). This function starts the MBROLA engine loads the requested diphone database and set all the internal

parameters to their default values. All the existing MBROLA databases are compatible with this application.

The stream provided by the external can be frozen (*Messages* inlet). It means that the phonetic/prosodic content stays in memory but the MBROLA engine stops the synthesis task.

Figure 9. Supported messages of the MaxMBROLA~ external object.



The MBROLA engine can also be stopped (*Messages* inlet). That function flushes the phonetic/prosodic content, stops the synthesis process and sets all the internal parameters to their default values. The diphone database remains loaded.

Fs inlet receives a floating point number. It controls the output sampling rate. Indeed, the original sampling rate depends on the database (16000Hz or 22050Hz). Linear interpolation is performed allowing the use of that external object with all possible sampling rates.

The inlets *Time*, *Pitch* and *Voice* each receive a floating point number. These values are respectively the time ratio (deviation of the reference speed of speech), the pitch ratio (deviation of the reference fundamental frequency of speech) and voice ratio (compression/dilation ratio of the spectrum width). For each inlet, 1.0 is the default value. The object doesn't transmit values lower than 0.01 (means "100 time lower than the default value").

2) Phonetic/prosodic stream processing

The requests for generating speech in the Max environment are described. All the following messages are sent into the *Messages* inlet.

A loading request allows to use a standard *.pho file (which include the list of phonemes to be produced and the target prosody) to perform synthesis. Examples are available together with MBROLA voices and complete explanations about standard SAMPA (Speech Assessment Methods Phonetic Alphabet). SAMPA is a machine-readable phonetic alphabet used in many speech synthesizers. (Cf. the SAMPA-page <http://www.phon.ucl.ac.uk/home/sampa/home.htm>).

We developed a function that directly accepts SAMPA streams inside Max messages to provide user control to interactive speech production. The standard SAMPA notation

has been modified to fit to the Max message structure. For example, the following stream:

```
phonemes & b 50 0 156 * a 150 100 204 #
```

begins by initializing the synthesizer, then produces a syllable /ba/ of 200 (50 + 150) milliseconds with a fundamental frequency increasing from 156Hz to 204Hz (two pitch points). Finally, it flushes the phoneme buffer.

D. Adding Text-to-Phoneme capabilities to MaxMBROLA

MaxMBROLA requires a phonemic specification as input just like it is used in mbrola .pho files, i.e. a transcription in SAMPA with optional information on duration and pitch. MaxMBROLA, just as mbrola, is not intended to be a fully fledged text-to-speech system. Anyway, it is obviously advantageous to combine it more directly with some kind of text-to-phoneme preprocessing in order to increase the flexibility of the system.

It was thus decided to use the text-to-phoneme capabilities provided by the TTS-system Mary (Schröder & Trouvain, 2003).

Mary is a Text-To-Speech system available for German and English. One of its attractive properties is that it offers full access to the results of intermediate processing steps. It provides an XML representation that contains not only the phonemes, their durations and pitch, but also a straightforward encoding of the full prosodic hierarchy which comprises phrases, words and syllables.

As there are applications of MaxMBROLA where the speech is to be synthesized syllable-wise, the latter information is most valuable.

A collection of simple Perl-scripts for parsing and converting Mary-XML format as well as standard mbrola .pho files to the input format required by MaxMBROLA was produced.

Max/MSP provides a "shell"-object which allows the execution of shell-commands, including piping, within a patch. This made the smooth integration of the text-to-phoneme processing rather straightforward.

As Mary is implemented as server-client architecture, as a special treat Mary was currently not installed locally but was accessed via Internet from within Max/MSP.

E. Gestural control of the Text-to-Speech system

Only one parameter, namely fundamental frequency (F0), was controlled by the glove in the MaxMbrola + mary text-to-Speech system. The phoneme stream and segment durations were computed by the TTS system. A flat pitch MBROLA signal was computed according to this data. Then F0 movements were computed by a PSOLA post-processing module receiving the flat MBROLA synthesized speech as input. F0 was modulated in real time, according to the distance between the glove and a fixed device on the table. This very simple control scheme was very effective. Very realistic and expressive prosodic variations were produced almost immediately because controlling F0 this way proved very intuitive.

V. REAL TIME CONTROL OF F0, DURATIONS AND INTENSITY IN A SYLLABLE BASED SPEECH SYNTHESIS SYSTEM USING MIDI-MBROLA

A. MIDI-MBROLA: The First MaxMBROLA-based MIDI Instrument

A Max/MSP musical instrument, called MIDI-MBROLA, has also been developed around the MaxMBROLA external object (D'Alessandro & al. 2005). This tool has a full MIDI compatible interface. MIDI *control changes* are used to modify the internal parameters of the MBROLA synthesizer. *Events* from a MIDI keyboard are used to compute the prosody, which is mixed with the phonetic content at the time of performance. As a standard module of the Max/MSP environment, the MIDI-MBROLA digital instrument automatically allows polyphony. Indeed, many voices can readily be synthesized simultaneously because the MBROLA synthesis doesn't utilize many CPU resources. It can also be compiled as a standalone application or as a VST instrument ("Virtual Studio Technology", a digital effect standard developed by Steinberg) instrument. That tool is publicly available.

B. Gestural control of MIDI-MBROLA

The MIDI-MBROLA instrument has been linked to the Roland keyboard and the three knobs of the Eobody Controller. The input text consisted of a syllabic sliced phonetic transcription of the speech utterance. Syllables were triggered by the keyboard. F0 was modulated by the keyboard and pitch-bend. Note that the keyboard has been divided in 1/3 of semitone between to adjacent keys. The Pitch Bend allowed for even smaller pitch excursions. The three knobs were controlling the overall speed, the mid-pitch and the vowel length. But it should be noticed that only the pitch control was effectively driving a parameter in real time whereas the three others were only sampled at syllables frequency (his means that once triggered a syllable was played with a given speed, without variation within the syllable). The configuration used is showed in Figure 9. With this configuration, the output speech had a singing character which sounded rather unnatural for speech. This was because the pitch variations were limited by the discrete nature of the keyboard.

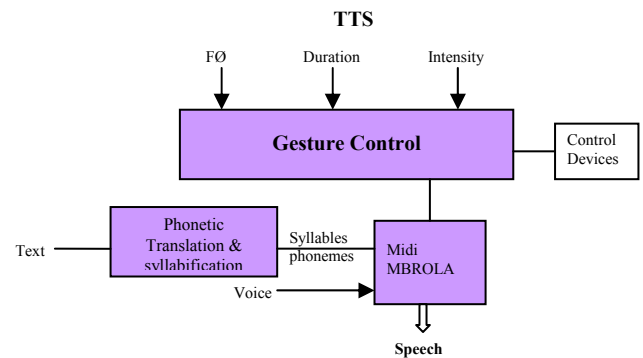
VI. FUJISAKI INTONATION MODELLING

Another strand of development dealt with the implementation of the Fujisaki model of intonation (Fujisak & Hirose, 1984) in the Pure Data environment. This model aims to take physical and physiological processes involved in the production of F0 into account. The main idea is to model the intonation contour by superimposing the results of two different processes. On the one hand there is the phrase component that models the phenomenon of slowly declining global pitch baseline throughout a prosodic phrase. The accent component, on the other hand, is responsible for modeling the local excursions in the F0 contour used for marking pitch accents.

Fujisaki's model has proved its descriptive adequacy in capturing F0 contours for a variety of different languages. As the input parameters of the model that have to be dynamically controlled can be basically reduced to the triggering of phrase commands, accent commands and their respective amplitudes, it seemed worthwhile to investigate its applicability in a real-time system.

An implementation of the Fujisaki in PureData was produced. In a first experiment the parameters where controlled by a MIDI-keyboard, where attack, release and velocity map quite straightforwardly to the timing and the amplitude of both accent- and phrase commands.

Figure 10. TTS Control Model



VII. DISCUSSION AND CONCLUSION

A. Summary of software produced.

Four different software projects have been produced during eNTERFACE:

1. the time-domain Baby Synthesizers. A LF model based vowel formant synthesizer, written in Pure Data, and mainly tested with keyboard, joystick and pedal-board real-time interfaces.
2. the spectral domain Baby synthesizer. A CALM model based vowel formant synthesizer, written in Max/MSP, and mainly tested with a digital glove real-time interface.
3. the Mary TTS in English with real-time intonation control, using a digital glove.
4. the MIDI-MBROLA speech synthesizer in French with a real-time control of intonation, duration and intensity using a keyboard with pitch bend.

B. Comparing patch programming Environments

Baby Synthesizers were developed using the real-time graphical environments Pure Data (PD) and Max/MSP. PD is an Open Source platform developed and maintained by Miller Puckette and includes code written by wide community of programmers. Max/MSP is commercial software developed by Cycling'74 company.

During this process we also tested some limits of these closely related platforms, and learnt lessons which we share.

Graphical environment

Being a commercial product, Max/MSP environment is better designed and user friendlier. However, simpler PD user

interface wasn't causing any problems in development process.

Stability

No stability issues with Max platform were encountered during the development. On the other hand, Pure Data programmers experienced several challenging problems, when some objects kept changing their behavior, disappearing and reappearing randomly. In general, stability issues were less serious for MacOS then for Windows platform; even system reboot didn't always help...

Richness

PD proved to be slightly more flexible when it came to coding more complex mathematical functions on sound wave in real time. Unlike Max/MSP, it allows a wide variety of mathematical operations to be performed in real-time directly on the sound signal with one very simple universal object. Similar operations had to be coded in C, compiled and imported to MAX/MSP.

Despite of the limitations mentioned above, both of these closely related environments proved to be suitable for sound processing applications of the kind we were developing.

C. Towards expressivity evaluation

Up to now no formal evaluation of the different variants of synthesizers has been performed. As a matter of fact, the evaluation of the "quality" of a speech synthesis system is not a trivial task in general, and is even more complicated when it comes to the evaluation of expressivity.

Usually synthesis systems are evaluated in terms of intelligibility and "naturalness". For the former there exist a number of established tests (Gibbon et al. 1997). Typically samples of isolated syllables or nonsense words are presented and it is possible to perform a quantitative evaluation of correctly perceived samples. When evaluating the "naturalness" of synthesized speech, an objective measure is less straightforward. In the simplest case, a comparison between two systems or between two variants of a system by forced preference choice can be performed. Another method is the rating of the "adequacy" of a synthesized sample for a given context. But again it is difficult to impossible to come up with an objective independent evaluation.

In the field of the synthesis of expressive speech, the predominant evaluation method is to synthesize sentences with neutral meaning and encode a small set of "basic" emotions (typically joy, fear, anger, surprise, sadness). Subjects are then asked to identify the emotional category.

A competing evaluation model is to use more subtle expressive categories: use test sentences with non-neutral semantics, and let again rate the adequacy of the sample for a given context.

In the context of the Speech-Conductor project it was only possible to perform an informal comparison of the two synthesizers that implemented glottal source models. At the current state the CALM based model gives much better "impression" than the time-domain model. On the other hand there are still a number of slight differences in the actual implementation of these two models; e.g. the differences in

the modeling of jitter and shimmer or the automatic superimposing of micro-prosodic variations, that have a strong impact on the perceived "quality" of the models.

A more interesting evaluation would be a rating test for the recognizability of perceptual voice quality measures such as laxness/tenseness, vocal effort etc. Though this would be probably a promising method of evaluating the current state, it is not easy to perform, as it would rely on the availability of independent "expert" listeners with a certain amount of phonetic experience.

In this context it would thus be interesting to further investigate whether it is possible to get reliable ratings on voice quality factors from so called "naive listeners".

For the MaxMBROLA system different evaluation methods have to be taken into account, as this is basically a classical diphone-synthesis system which allows for the real-time control of prosodic features, most prominently pitch. Thus the evaluation methods used for "normal" concatenative synthesis systems could easily applied. One of the peculiarities of this system is that inevitable the virtuosity of the person "conducting" the synthesizer is a strong factor in the quality of the output.

A straightforward evaluation would be a rating test of different input devices (e.g. Data Glove vs. Keyboard), but apart from the "human factor", currently still too many differences in the underlying synthesis scenarios exist to allow a real comparison.

D. Conclusion

Devices:

The glove performed much better than the keyboard or joysticks for controlling intonation and expressivity. However, the tested glove model had some performance limitations (it proved too slow for real time). However, the glove wasn't tested for its capacity to reproduce the intended gesture precisely and reliably.

Keyboard on the contrary allows for exact reproducibility of gestures. When combined with TTS synthesizers the produced speech had somewhat singing quality, as pitch changes are directly linked to syllable onsets.

Synthesizers:

In general, voice source models produced much more expressive vocal utterances than TTS models. For TTS, better results were reached when speech was generated using pre-computed segment durations and intensity and we only controlled F_0 . So, surprisingly, less control can in some situations yield better results. In any case, it's clear that to add a real expressivity, flexible control of all of the voice source parameters is needed.

To our best knowledge, this project was the first attempt to implement real-time system of gestural control of expressive speech. The results proved really encouraging, and opened a new avenue for expressive speech synthesis research.

ACKNOWLEDGEMENTS

Hannes Pirker states that his research is carried out within the Network of Excellence Humaine (Contract No. 507422) that is funded by the European Union's Sixth Framework Programme with support from the Austrian Funds for Research and Technology Promotion for Industry (FFF 808818/2970 KA/SA). This publication reflects only the author's views. The European Union is not liable for any use that may be made of the information contained herein.

REFERENCES

- (Bozkurt et al., 2005) B. Bozkurt, B. Doval, C. D'Alessandro, and T. Dutoit, "Zeros of Z-Transform Representation With Application to Source-Filter Separation in Speech" IEEE Signal Processing Letters, Vol. 12, No. 4, April 2005, p 344-347
- (d'Alessandro, 2005) C.d'Alessandro, "Voice source parameters and prosodic" analysis, in Methods in Experimental prosody research, Mouton de Gruyter (in press)
- (d'Alessandro & Doval, 2003) C. d'Alessandro, B. Doval, "Voice quality modification for emotional speech synthesis", Proc. of Eurospeech 2003, Genève, Suisse, pp. 1653-1656
- (D'Alessandro et al., 2005) N. D'Alessandro, B. Bozkurt, T. Dutoit, R. Sebbe, 2005, "MaxMBROLA: A Max/MSP MBROLA-Based Tool for Real-Time Voice Synthesis", Proceedings of the EUSIPCO'05 Conference, September 4-8, 2005, Antalya (Turkey).
- (Doval & d'Alessandro, 1999) B. Doval, C. d'Alessandro, 1999. *The spectrum of glottal flow models*. Notes et Documents LIMSI 99-07, 22p.
- (Doval & d'Alessandro, 1997) B. Doval and C. d'Alessandro. *Spectral correlates of glottal waveform models: an analytic study*. In International Conference on Acoustics, Speech and Signal Processing, ICASSP 97, pages 446--452, Munich, avril 1997. Institute of Electronics and Electrical Engineers
- (Doval et al., 2003) B. Doval, C. d'Alessandro, and N. Henrich, "The voice source as a causal/anticausal linear filter," in *Proc. ISCA ITRW VOQUAL 2003*, Geneva, Switzerland, Aug. 2003, pp. 15–19
- (Dutoit et al., 1996) T. Dutoit, V. Pagel, N. Pierret, F. Bataille and O. van der Vrecken, "The MBROLA project: towards a set of high quality speech synthesizers free of use for non commercial purposes" Proc ICSLP, Philadelphia, pp. 1393-1396, 1996.
- (Fant et al., 1985) Fant G., Liljencrants J. and Lin Q. (1985) "A four-parameter model of glottal flow". STL-QPSR 4, pp. 1-13.
- (Fant, 1995) G. Fant, "The LF-model revisited. Transformation and frequency domain analysis," *Speech Trans. Lab. Quarterly Rep., Royal Inst. of Tech. Stockholm*, vol. 2-3, pp. 121-156, 1995.
- (Fels, S. 1994) Fels, "Glove talk II: Mapping hand gestures to speech using neural networks," Ph.D. dissertation, Univ. Toronto, Toronto, ON, Canada, 1994.
- (Dutoit, 1997) Dutoit T. An Introduction to Text-To-Speech Synthesis. Kluwer Academic Publishers, 1997.
- (Fujisaki & Hirose, 1984) H. Fujisaki, K. Hirose Analysis of voice fundamental frequency contours for declarative sentences of Japanese Journal of Acoustic Society. Jpn. (E) 5, 4. 1984
- (Gibbon et al., 1997) Gibbon, D., Moore, R. & Winsky, R. (Eds) *Eagles handbook of Standards and Resources for Spoken Language Systems* (1997) Mouton de Gruyter
- (Henrich et al. 2002) N. Henrich, C. d'Alessandro, B. doval. "Glottal flow models: waveforms, spectra and physical measurements". Proc. Forum Acusticum 2002, Séville 2002
- (MIDI, 1983) "MIDI musical instrument digital interface specification 1.0," Int. MIDI Assoc., North Hollywood, CA, 1983.
- (Schröder, 2004) M. Schröder "Speech and emotion research", *Phonus*, Nr 7, june 2004, ISSN 0949-1791, Saarbrücken
- (Schröder & Trouvain, 2003) M. Schröder & J. Trouvain (2003). The German Text-to-Speech Synthesis System MARY: A Tool for Research, Development and Teaching. *International Journal of Speech Technology*, 6, pp. 365-377.
- (Zicarelli et al., 2004b) Zicarelli, G. Taylor, J. K. Clayton, J. and R. Dudas, MSP 4.3 Reference Manual. and Max 4.3 Reference Manual. Cycling'74/Ircam, 1990-2004.
- (Wanderley & Depalle; 2004) M. Wanderley and P. Depalle, "Gestural Control of Sound Synthesis", *Proc. of the IEEE*, 92, 2004, p. 632-644.
- <http://mary.dfki.de>
- <http://tcts.fpms.ac.be/synthesis/maxmbrola/>
- <http://www.disc2.dk/tools/SGsurvey.html>

MASTER-PIECE: A Multimodal (Gesture+Speech) Interface for 3D Model Search and Retrieval Integrated in a Virtual Assembly Application

Konstantinos Moustakas¹, Dimitrios Tzovaras¹, Sebastien Carbini², Olivier Bernier², Jean Emmanuel Viallet², Stephan Raidt³, Matei Mancas⁴, Mariella Dimiccoli⁵, Enver Yagci⁶, Serdar Balci⁶, Eloisa Ibanez Leon⁷.

Abstract—The present report presents the framework and the results of Project 7: "A Multimodal (Gesture+Speech) Interface for 3D Model Search and Retrieval Integrated in a Virtual Assembly Application", which has been developed during the eNTERFACE-2005 summer workshop in the context of the SIMILAR NoE. The "MASTER-PIECE" (Multimodal Assembly with SIMILAR Technologies from European Research utilizing a Personal Interface in an Enhanced Collaborative Environment) project aims at the generation of a multimodal interface using gesture and speech in order to manipulate a virtual assembly application. Besides assembling mechanical objects, the user is capable to perform 3D model content based search in a database of 3D objects using as query model a scene object. Finally, to deal with cases where no query model is available, a sketch based approach is proposed which results in the manual approximate generation of the query model. More specifically, the user can draw a specific number of primitive objects by moving his/her hands and then process-combine them so as to build more complex shapes, which are finally used as query models. Experimental results illustrate that the proposed scheme enhances significantly the realism of the interaction, while using the sketch-based approach the user can search for 3D objects in the database without the need of an initial query object, which is the case in most state of the art approaches.

Index Terms—Multimodal interface, virtual assembly, 3D search, gesture, speech, sketch recognition.

I. INTRODUCTION

During the latest years there has been an increasing interest in the Human-Computer Interaction society for multimodal interfaces. Since Sutherland's SketchPad in 1961 or Xerox' Alto in 1973, computer users have long been acquainted with more than the traditional keyboard to interact with a system.

This report, as well as the source code for the software developed during the project, is available online from the eNTERFACE'05 web site: www.enterface.net.

1: Informatics and Telematics Institute, Centre for Research and Technology Hellas, Thessaloniki, Greece, e-mail: moustak@iti.gr, tzovaras@iti.gr

2: France Telecom R&D, Lannion, France, e-mail: sebastien.carbini@rd.francetelecom.com, olivier.bernier@francetelecom.com, jeanemmanuel.viallet@rd.francetelecom.com

3: INP-Grenoble, Grenoble, France, e-mail: Stephan.Raidt@icp.inpg.fr

4: Faculte Polytechnique de Mons, Mons, Belgium, e-mail: matei.mancas@tcts.fpm.ac.be

5: Universitat Polytechnica de Catalunya, Barcelona, Spain, e-mail: mariella@gps.tsc.upc.es

6: Bogazici University, Istanbul, Turkey, e-mail: enveryagci@yahoo.com, serdar.balci@boun.edu.tr

7: Technical University of Madrid, Madrid, Spain, e-mail: eloisa.i@caramail.com

More recently with the desire of increased productivity, of seamless interaction and immersion, of e-inclusion of people with disabilities, as well as with the progress in fields such as multimedia/multimodal signal analysis and human-computer interaction, multimodal interaction has emerged as a very active field of research (e.g. [1] [2]).

Multimodal interfaces are those encompassing more than the traditional keyboard and mouse. Natural input modes are put to use (e.g. [3], [4]), such as voice, gestures and body movement, haptic interaction, facial expressions and more recently physiological signals. As described in [5] multimodal interfaces should follow several guiding principles: multiple modalities that operate in different spaces need to share a common interaction space and to be synchronized; multimodal interaction should be predictable and not unnecessarily complex, and should degrade gracefully for instance by providing for modality switching; finally multimodal interfaces should adapt to user's needs, abilities, environment.

A key aspect in multimodal interfaces is also the integration of information from several different modalities in order to extract high-level information non-verbally conveyed by users. Such high-level information can be related to expressive, emotional content the user wants to communicate. In this framework, gesture has a relevant role as a primary non-verbal conveyor of expressive, emotional information. Research on gesture analysis, processing, and synthesis has received a growing interest from the scientific community in recent years and demonstrated its paramount importance for human machine interaction (see for example the Gesture Workshop series of conferences started in 1996 and since then continuously growing in number and quality of contributions; a selection of revised papers from the last workshop can be found in [6]).

MASTER-PIECE integrates gesture and speech modalities into a designer and assembly application so as to increase the immersion of the user and to provide a physical interface and easier tools for design, than the mouse and the keyboard. Moreover, the user is capable of generating simple 3D objects and search for similar 3D content in a database, which is nowadays another very challenging research topic.

In particular, search and retrieval of 3D objects has application branches in numerous areas like recognition in computer vision and mechanical engineering, content-based search in e-commerce and edutainment applications etc. [7], [8], [9]. These application fields will expand in the near future, since

the 3D model databases grow rapidly due to the improved scanning hardware and modeling software that have been recently developed.

The difficulties of expressing multimedia and especially three dimensional content via text-based descriptors reduces the performance of the text-based search engines to retrieve the desired multimedia content efficiently and effectively. To resolve this problem, 3D content-based search and retrieval (S&R) has drawn a lot of attention in the recent years.

However, the visualization and processing of 3D models are much more complicated than those of simple multimedia data [10], [11], [12]. The major difference lies in the fact that 3D models can have arbitrary topologies and cannot be easily parameterized using a standard template, which is the case for images. Moreover, there can be many different models of representing them, i.e. indexed facets, voxel models etc. Finally, processing 3D data is much more computationally intensive, than processing media of lower dimension, and often requires very large amounts of memory.

Many researchers worldwide are currently developing 3D model recognition schemes. A number of approaches exist in which 3D models are compared by means of measures of similarity of their 2D views [13], [14]. More direct 3D model search methods focus on registration, recognition, and pairwise matching of surface meshes [15], [16], [17]. Unfortunately, these methods usually require a computational costly search to find pairwise correspondences during matching.

Significant work has also been done in matching 3D models using geometric characteristics, where initial configurations are derived from conceptual knowledge about the setup of the acquisition of the 3D scene [18] or found automatically by extracting features such as curvature or edges [19]. When correspondences between the two objects to be matched are unknown, the registration problem, which in general is not well posed, may approximately be solved by the iterative closest point (ICP) algorithm [20]. In the absence of a priori knowledge or robust features, the ICP algorithm starts with one unique or, preferably, multiple different initial configurations [21]. In [22], a framework is presented for analyzing the subspace of the complete configuration space so as to force the ICP algorithm to converge to the global minimum. The method is evaluated experimentally for a number of real 3D objects.

A typical S&R system, like the aforementioned ones, evaluates the similarities between query and target objects according to low-level geometric features. However, the requirement of a query model to search by example often reduces the applicability of an S&R platform, since in many cases the user knows what kind of object he wants to retrieve but does not have a 3D model to use as query.

Imagine the following use case: The user of a virtual assembly application is trying to assemble an engine of its spare parts. He inserts some rigid parts into the virtual scene and places them in the correct position. At one point he needs to find a piston and assemble it to the engine. In this case, he has to manually search in the database to find the piston. It would be faster and much more easier if the user had the capability of sketching the outline of the piston using specific

gestures combined with speech in order to perform the search. In the context of this project the integration of speech and gestures for the generation of the query model is addressed. Speech commands are used for performing specific actions, while gesture recognition is used to draw a sketch of the object and to manipulate the scene objects in the 3D space. The system is also capable of deforming objects and combine them so as to build more complex structures.

The rest of the document is organized as follows. Section II presents the general concepts of the developed application framework. In Section III the virtual assembly application and the 3D search engine are briefly described. Section IV describes the developed multimodal interface to the application and analyzes in detail all techniques used for the generation of the query model using sketches, while Section V presents the action verification module, which consists of a talking head. Finally, in Sections VI and VII two of the performed experiments are described and conclusions are drawn respectively.

II. APPLICATION FRAMEWORK

The developed application framework is a 3D assembly-designer interface. The user is capable to:

- Manipulate 3D objects in a 3D environment, including translation, rotation, scaling, etc.
- Assembly mechanical objects from their spare parts.
- Import 3D primitive objects using sketches.
- Manipulate and deform the primitive objects so as to generate more complex structures.

Unfortunately, an interface, like the predescribed one, is very difficult to use with standard keyboard-mouse input devices. The major problem stems from the fact that 3D actions can not be easily reproduced using 2D input devices.

The aim of presented framework is to add physical means of interaction between the user and the application and to overcome the need of the transition between the 2D input devices (mouse, etc.) and the 3D virtual environment. In particular, the developed multimodal interface consists of the modules:

- Speech recognition for specific commands.
- Gesture recognition to efficiently handle 3D objects using 3D motions of the hands.
- Recognition of sketches.
- Primitive models import and manipulation using gestures.
- Deformation of objects using gestures.

Finally, Figure 1 illustrates a block diagram of the developed multimodal interface

III. BASIC MODULES

A. Virtual assembly application

The virtual assembly application is a graphical 3D interface for performing assembly of mechanical objects from their spare parts as illustrated in Figures 2a and 2b.

It has been initially developed to be used with haptic gloves and it allows the user to:

- Assembly a mechanical object from its spare parts.
- Grasp and manipulate objects using haptic gloves.

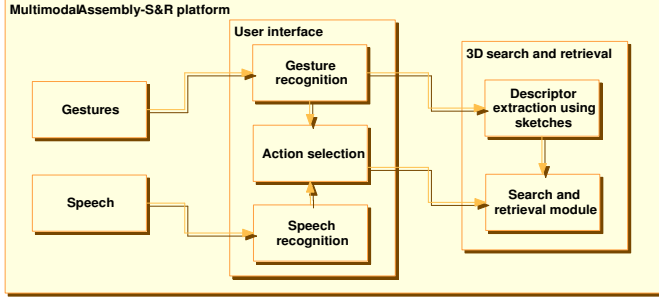


Fig. 1. Block diagram of the developed multimodal application

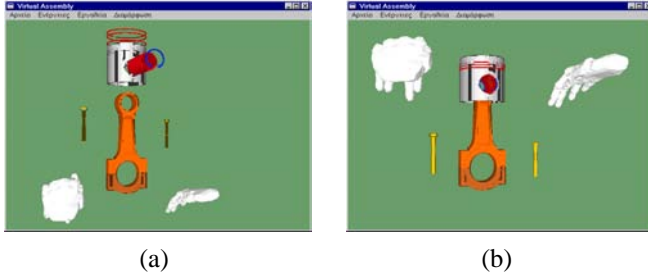


Fig. 2. Virtual assembly application

- Author assembly scenarios using the authoring tool.

The user is also capable of assembling parts of an object and record the assembly process for post-processing. The assembly procedure can be done using one or two hands (i.e. one or two haptic VR gloves CyberTouch or CyberGrasp). A position tracker (MotionStar Wireless Tracker of Ascension Technologies Inc. (2000)) with one or two position sensors installed is used to detect the position and orientation of the user hands in the space.

Another element of the application is the Virtual Reality (VR) agents module. VR agents are sophisticated software components with the capability to control a dynamic virtual environment and take actions based on a set of aims and rules. There are two kinds of agents implemented in the VR Assembly environment: a) the snap agent and b) the tool agents (a screwdriver and a wrench).

The snap agent is responsible to decide when two parts in the scene must connect. The aim of that agent is to place the components of the assembly in the correct position and to allow two or more components to construct a new larger component that can act like any other component. The rule that snap agent uses to connect two objects is a distance threshold and a "first contact rule". The "first contact rule" detects which sides of the objects collide first (using bounding boxes). If the colliding sides are valid, i.e. the objects are approaching from the sides that can snap, then the distance from the current position to the snapping position is calculated. When this distance is smaller than the distance threshold the objects snap to each other. The distance threshold used depends on the radius of the smaller bounding sphere of the objects.

The tools are components with the capability to control objects in the dynamic virtual environment. The tools aim to increase the immersion of the user in the VE. Unlike the snap

agent, which is always active, the tools need to be activated by the user. Tools provide constraint to the object movement and allow the user do construction tasks. The virtual tools have the potential to increase user productivity by performing tasks on behalf of the user and increase the immersion of the user in the virtual environment. Furthermore use of virtual tools during assembly in the VE aids the user to detect possible construction difficulties related to the position and shape of the tools.

During the workshop the application has been extended to be used with a standard mouse-keyboard interface and also using the multimodal gesture-speech interface.

B. 3D content-based search engine

Using the 3D content-based search engine 3D objects can be retrieved from a database using another 3D object as query. Then the engine retrieves the most similar, in terms of a distance metric between their descriptor vectors, objects to the query model. The algorithm for extracting the geometrical descriptor of the object is briefly described in the sequel.

The extracted descriptors are rotation invariant. In particular, the object is initially normalized in terms of translation and scaling, i.e. it is translated to the center of the coordinate system, and is scaled uniformly so that the coordinates of all its vertices lie in the interval $[0, 1]$. Next, N_c concentric spheres are built centered at the origin of the coordinate system. Each sphere is built using tessellation of a normal icosahedron so that the vertices over its surface are uniformly distributed. In the experiments 20 concentric spheres of 16002 vertices are used. For each sphere the discrete 3D signal $F(r_s, \theta_i, \phi_i)$ is assumed, where i is the index of the sphere vertices. The values of function $F(r_s, \theta_i, \phi_i)$ are calculated using the Spherical Trace Transform (STT) [23].

The extraction of the final descriptor vectors, which is used for the matching algorithm, is achieved by applying the spherical functionals T , as described in [23], to the initial features $F(r_s, \theta_i, \phi_i)$ generated from the STT. The spherical functionals for each concentric sphere ρ are summarized below:

$$1. T_1(F) = \max\{F(r_s, \theta_i, \phi_i)\} \quad (1)$$

$$2. T_2(F) = \sum_{j=1}^{N_s} |F'(r_s, \theta_i, \phi_i)| \quad (2)$$

$$3. T_3(F) = \sum_{j=1}^{N_s} F(r_s, \theta_i, \phi_i) \quad (3)$$

$$4. T_4(F) = \max\{F(r_s, \theta_i, \phi_i)\} - \min\{F(r_s, \theta_i, \phi_i)\} \quad (4)$$

$$5. T_l(F) = A_l^2 = \sum_m \alpha_{lm} \quad (5)$$

where N_s is the total number of sampled points ($\eta_j, j = 1, \dots, N_s$) at each concentric sphere, $l = 0, \dots, L$ and $-l < m < l$. α_{lm} are the expansion coefficients of the Spherical Fourier Transform [24]:

$$\alpha_{lm} = \sum_{i=1}^{N_s} F(r_s, \theta_i, \phi_i) Y_{lm}(\eta_i) \frac{4\pi}{N_s} \quad (6)$$

where $Y_{lm}(\eta_i)$ corresponds to the spherical harmonic function, which is defined through:

$$Y_l^m(\theta, \phi) = k_{l,m} P_l^m(\cos \theta) e^{jm\phi} \quad (7)$$

where P_l^m is the associated Legendre polynomial of degree l and order m , $k_{l,m}$ a normalization constant and j the imaginary unit.

The quantities A_l^2 are invariant to any rotation of the 3D model. Choosing a sufficiently large number of L coefficients of the Spherical Fourier Transform, a total number of $L + 4$ spherical functionals is used for each concentric sphere.

Finally, the descriptor vectors $\mathbf{D}(l)$ are created, where $l = 0, \dots, (L + 4)N_c$ is the total number of descriptors and N_c is the number of concentric spheres. In the experiments described in the sequel, $L = 26$ and $N_c = 20$ were chosen.

Figure 3 depicts the retrieved objects using as query the first model of each column.

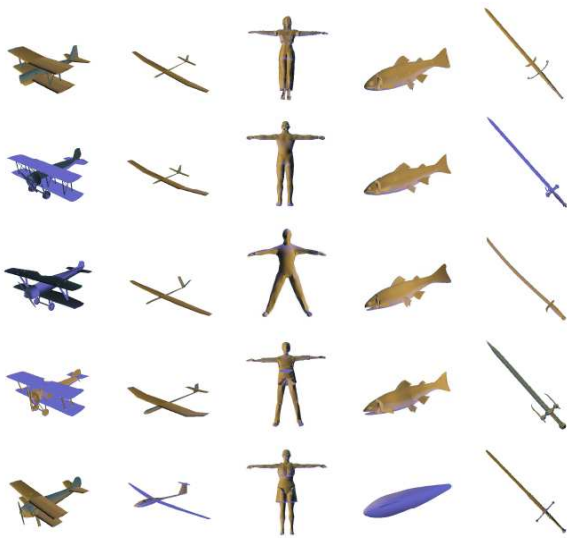


Fig. 3. 3D search results. The first row corresponds to the query object.

IV. MULTIMODAL INTERFACE TO THE APPLICATION

The developed multimodal interface to the application consists of two major parts. The interaction unit, which consists of the gesture-speech interface that enables the user to interact with the platform using only gesture and speech and the sketch-based query model generation system for the manual design of the model to search for. In the rest of this Section technical details about the algorithms used for the multimodal interface are presented.

A. Combined Gesture-Speech interface

1) *Body parts detection and tracking:* Most people instinctively use the eye-tip of the finger line to point at a target [25]. This convention is used in the present framework to estimate the pointing direction of the user [26], [25]. More precisely, the pointing direction is approximated by the head-hand axis.

Head and hands are detected and tracked [27] (Figure 4-left). Their positions in the 3D space are given by a stereo



Fig. 4. Left: camera image (rectangle: head, circle: hand1, cross: hand2). Right: observations assigned to one of the four models depending on their probabilities (blue: head, red: hand1, green: hand2, grey: discard, white: pixels ignored in EM).

camera. The face of the user is automatically detected by a neural network more precisely described in [28]. The hands are detected as skin coloured moving zone in front of a vertical plane passing through the head, triggering pointing gesture recognition.

The first detected hand is tagged as “pointing hand” and the second detected hand is tagged as “control hand”. The system can be used by right-handed persons as well as by left-handed persons (predominant hand is generally used to point) without differentiating explicitly right hand from left hand.

Once detected, the body parts (head and hands) are simultaneously tracked until tracking failure is automatically detected: then the detection for the lost part is re-triggered. The tracking process aims at explaining each new image by a statistical model with the EM algorithm [27]. The statistical model is composed of a colour histogram and a 3D spatial Gaussian function for each tracked body part (Figure 4).

2) *Gesture interpretation:* Gesture recognition is triggered by speech commands. The axis obtained from the first hand and the head 3D positions is used to compute pointed direction. The pointed direction is used when the user utters “selection” to select the pointed 3D object. Once selected, if the user utters “move”, the object keeps moving to the pointed direction until the user utters “O.K.”. When the word “rotate” is uttered, the current hand angles are taken as reference angles and current main axis of the object as reference axis. Then, until uttering “O.K.”, the object rotates around its center of mass following the user’s hands rotation according to the spherical coordinates (alpha and beta in Figure 5). When the user utters “scale”, the current 3D distance between the hands is taken as reference value and current size of the object as reference size. Then, until uttering “O.K.”, the object is continuously resized proportionately to the distance between hands (when distance between hands is two time higher than reference distance, the object is two time bigger than its reference size).

3) *Speech recognition:* To recognize speech commands, the speech signal is linearly sampled at 8 kHz in 16 bits. MFCC (*Mel Frequency Cepstrum Coefficients*) coefficients are computed, each 16 ms, on 32 ms signal frames. The recognition system uses the frame energy, 8 cepstral coefficients and an estimation of the first and second order derivatives of the speech signal. Thus, the observation vector has 27 dimensions.

The decoding system uses Hidden Markov Models. The

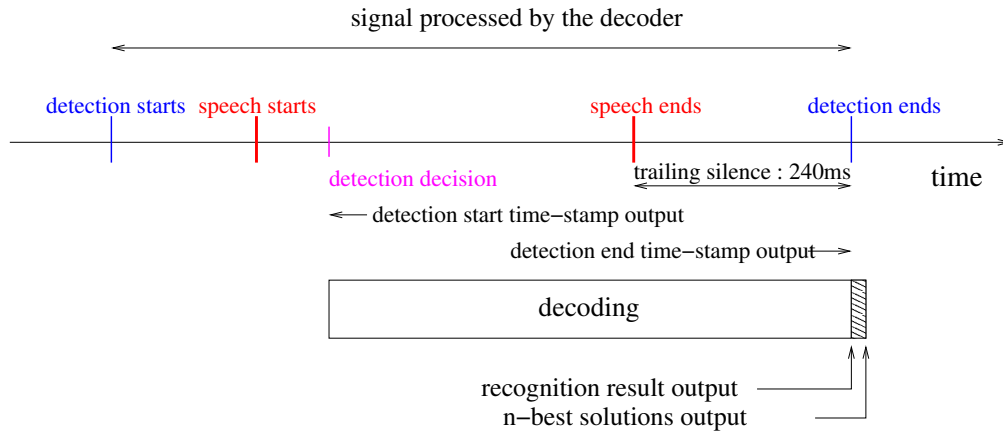


Fig. 6. Different times related to speech recognition. Speech recognition is only available after a delay following speech signal.

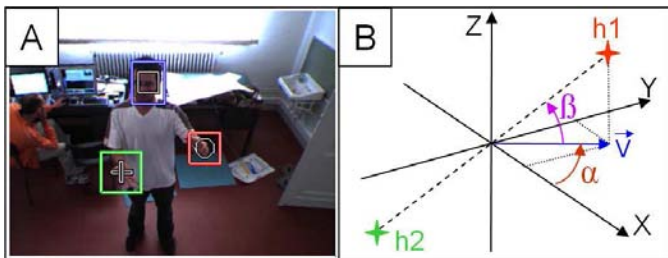


Fig. 5. (A): A user rotating a 3D object with his hands (blue: head, red: hand1, green: hand2). (B): the two hands h1 and h2 are defined by two angles α and β in the spherical coordinates.

recognized sentences syntax is described in a grammar. The used vocabulary consists of 50 words. Dependent on the context, each word is obtained by phonetic units concatenation named allophones [29]. The system outputs the n-best results [30].

A noise/speech detector component filters the input signal to provide the decoder only with speech signal surrounded by silent frames. Beginning of detection is not causal. Detection component provides several frames which precede the speech detection decision. But as the decoder is faster than real time, it recovers from the non-causality of the detection component.

To detect end of speech, some consecutive silent frames must be observed. These frames are sent to the decoder. The best solution can be provided as soon as the last frame has been received. Computing the n-best solutions generate a negligible lag compared to the lag due to the silent frames. The number of frames to detect the end of speech is a parameter of the noise/speech component set to 15. The lag between the end of speech and the result of the recognition is thus 240 ms, since frames are grabbed each 16 ms. These times include the start and end silent frames which differ from the times of start and end of speech. These former can be computed from the noise-speech parameters. All the times constraints are summarized in Figure 6.

B. Sketch-based query interface

In this section the sketch recognition system will be described. Sketches are acquired via the gesture recognition mod-

ule described in Section IV-A. The algorithms are designed only to recognize three basic shapes: square (or rectangle), circle and triangle. Figure 7 shows examples of traces obtained after acquisition.

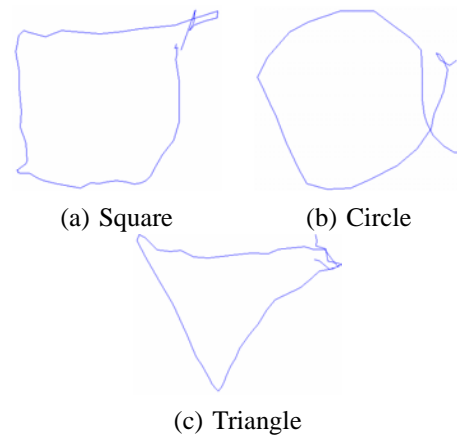


Fig. 7. Examples of a square, a circle and a triangle after trace acquisition

It is obvious that these traces are far from perfection and often contain noisy points at the beginning and the end of the trace. Moreover, the point sets created by hand gesture recognition are generally distorted, while the sampling density is in general not constant.

The shape recognition systems can be grouped into two main categories. The first one could be described as feature based or rule based. Shapes are grouped using some human recognition based characteristic features as the number of corners, the number of parallel sides, etc. The second group consists of the model-based methods, which are based on fitting the different possible shapes on the trace to be recognized and to select the shape providing the highest correlation level with the trace to recognize.

After describing the sketch denoising, the sketch recognition algorithms will be presented. Finally, possible extension of the sketch recognition system will be discussed.

1) *Sketch denoising*: As illustrated in Figure 7, some undesirable points occur in the acquired trajectory, especially at the beginning and the end of the trace. These points have to

be discarded to improve the results. Several approaches were designed and tested, both using directly the acquired points or transforming them into images. The result (discarding of noisy points) was evaluated in terms of correct recognition rate, robustness to noise, etc. and the main advantages and drawbacks are discussed at the end of this section. The sketch denoising methods can be divided into two more categories: those that only discard the meaningless points and those which also enhance and simplify the shape in order to make shape recognition more robust.

Method 1: Polar transform denoising

The first method discards some points considered as noise, which lie usually at the beginning and end of the drawing as can be seen in Figure 8. Points that follow in time according to their time stamp, but that do not correspond to a higher value of angle in the polar coordinate description, are deleted. This algorithm avoids also line crossings of the start and end points of the drawing.

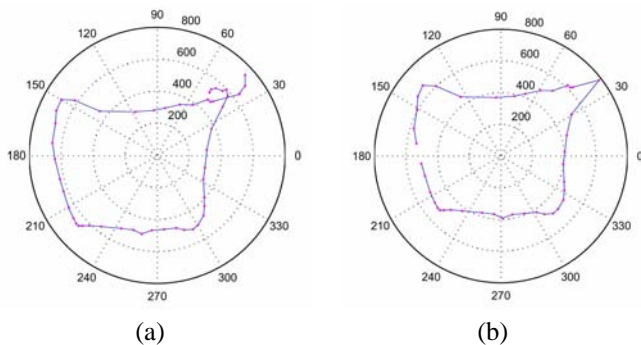


Fig. 8. Example of a square, before and after noisy points discarding using method 1

The main problems of this method are that we need to know which is the direction of the trace and which corner is the first one. So, extra algorithms are implemented to check that efficiently and aid the method.

Method 2: Convex denoising

The second method searches for intersections in the tracked trajectory of the hand. The first and last points are investigated whether there is any cross section between any line segments. If the result is positive, the colliding edges are merged and the remaining parts of the trajectory (i.e. the line segments that do not belong to the closed contour) are discarded. Results are shown in Figure 9.

This method works well on all closed figures. If a figure is not closed as the example shown in Figure 10, it will fail.

Method 3: Statistical denoising

This method uses two statistical thresholds based on mean distance between two consecutive points. The first T_1 is used to remove small irregularities. In order to decide if N consecutive points constitute a small irregularity or not, a threshold is compared with the geodesic distance between the latest $N+2$ points. If the geodesic distance is bigger than a threshold value, which represents the weighted average distance between N

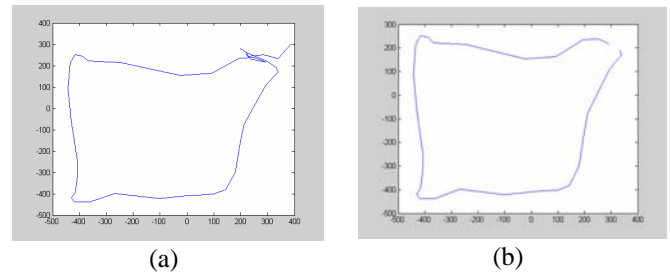


Fig. 9. Example of a square, before and after noisy points discarding using method 2

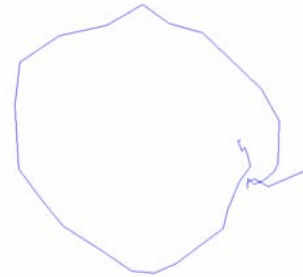


Fig. 10. Example of a circle, which is not closed

points, we consider these points as noise and delete them.

The second threshold T_2 is used to decide about the position where the trajectory should be closed so as to generate a closed contour. If there exists an intersection it is easy to decide about these position, as described in the previous methods. This method handles also cases of open trajectories. In particular, if the distance of the point considered, from at least one of the preceding points is smaller than T_2 these points are connected and the rest of the points outside the closed contour are discarded.

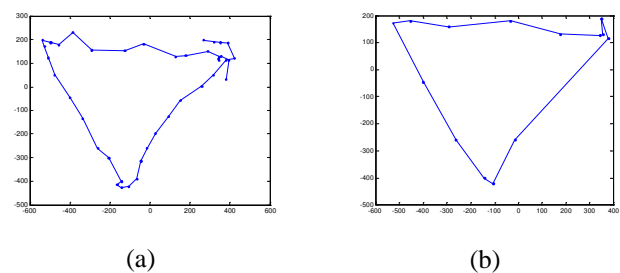


Fig. 11. Example of a square, before and after noisy points discarding using method 3

In order to avoid connection of consecutive points and immediate failure of the algorithm, the preceding points of the considered one that are closer to it, in terms of geodesic distance, than T_1 are not considered for this test. Figure 11 illustrates results of this method.

Method 4: Image-based denoising

The fourth method converts the initial data set into an image by placing the linked trajectory points into an image raster. Then, image processing techniques are applied to process the traces. First of all a simple test is applied to test if the trace

is closed or not just by recursively evaluating the filled area of the trajectory: if the area is exactly of the same size as the image the trace is open, but if this area is smaller than the image, that means our trace was already closed. This method closes the trace if it was open before further processing.

In a second step the shape size is normalized and morphological opening is used in order to get rid of the meaningless points and to smooth the traces. As we previously normalized the shape size, it is possible to find a fixed kernel size for the morphological opening, which works better according to final visual check. Some results are shown in Figure 12, and this method worked well on all our experiments. Smoother shapes are obtained without the noisy points, which usually exist at the beginning and the end of the traces. This method works with both close or opened traces and it preserve the main features of the traces (corners, parallel sides, roundness). A possible drawback is the fact that filling big images could be a little long, but this problem could be solved by normalizing all traces at smaller sizes. Another drawback could be the fact that the point set is transformed into an image, so it would be better to use a recognition method also done in the image space. But it is still possible to convert the denoised image into a set of points in order to use a recognition method working directly with the set of points.

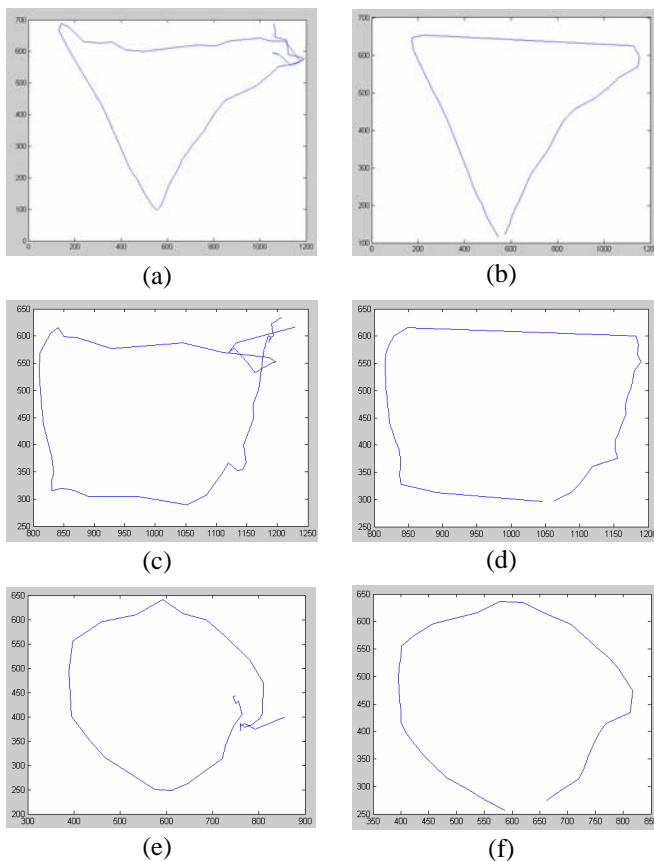


Fig. 12. Example of a triangle, square and circle, before (first column) and after (second column) noisy points discarding using method 4

2) *Sketch recognition*: The aim of the sketch recognition system is to recognize the shape of the trace and its main features (corners, center, radius, etc.) in order to draw then noiseless, squares, triangles and circles using the extracted features. The primitive-object vocabulary used demands the recognition of only 3 basic 2D shapes (triangle, circle or square). We tested several sketch recognition methods both in points set or image space.

Method 1: Polar transform recognition

In this method polar coordinates are used. In the first step of the recognition process the center of the shape is calculated as the mean of the X and Y coordinates. Measured points are usually not equally distributed so this center does not coincide with the real center of the shape but this problem is eliminated by using resampling, to get equally spaced sample points, and a denoising method as previously described.

At the beginning the distance between every point and the center is calculated and plotted into a "Distance-Angle" graph as illustrated in Figure 13. Theoretically, triangles should have three maxima/minima in this system, squares should have four (because of the three or four corners which are far away from the center) and circles may have many but very local maxima. For the triangle these peaks are equally spaced at 120° and at 90° in the case of the square. This interrelation is exploited for the recognition process. The detection of peaks and their angular distance gives sufficient information for the recognition comparing the calculated values to threshold values. The corresponding display of the example square in Figure 13a shows four distinct peaks. Ideally, these four peaks will be equally spaced and have the same amplitude, which is not the case here, due to noise, not exact sketching, etc.

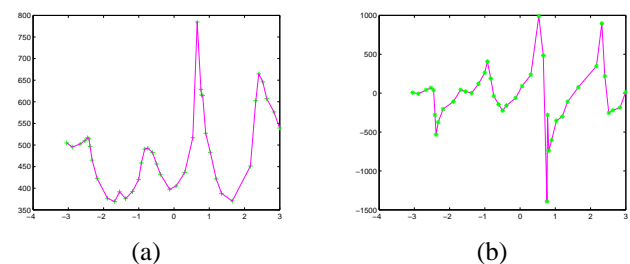


Fig. 13. a) Cartesian display of the polar coordinates (distance from center on the Y-axis, angle on the X-axis), b) Result of differentiation of the distance rho

For making the system more robust it's been decided to differentiate the distance to the centre (ρ) with respect to the angle (θ). This yields zero crossings where the peaks are with positive peak before and negative peaks after the zero crossing. This should allow easier localization of the peaks. The decision criteria of number of peaks and their distance remain. The circle should neither show distinct peaks nor equal distance between them, whereas their number may be very different. The result of differentiation is shown in Figure 13b.

Method 2: Vector-based recognition

The vector-based method takes advantage of the fact that the shapes to recognize have specific features that can be used for their recognition. I.e. lines consecutive points of a rectangle will have two main directions, of a triangle three, and for the circle many. The following assumption are considered in this algorithm:

- 1) Every rectangle consists of two sets of parallel lines.
- 2) Every triangle consists of three lines.
- 3) Circles cannot be separated into a specific finite number of lines
- 4) We have three different classes; triangle rectangle and circle and any given point set belongs to one of them.

This scheme is based on building a histogram of the direction of the line segments between two consecutive points, i.e. a histogram with the polar coordinate angles of each line segment. If the shape is a square then the histogram has two clearly defined maxima, corresponding to the directions of the two sets of parallel lines. If it is a triangle it should have three clearly defined maxima, while if it is a circle it has almost a uniform distribution. Using these features the task of classifying each point set into one of the three classes becomes trivial.

Method 3: Image-based recognition

Another idea is to use the image space. After transposing the point set into an image, as previously described, the denoising method 4 is used to obtain proper filled shapes. Then, the bounding boxes are computed. Next, by comparing the area of the shapes to the bounding box area we can recognize them using the following procedure.

If the area is close to the bounding box the point set should represent a rectangle. If it is smaller it should correspond to a circle and finally if it is even smaller, it should be a triangle. An example is illustrated in Figure 14. At the left the shape bounding box and a perfect circle (an ellipse here) fitting into it is depicted. The right image illustrates the shape to classify. By comparing its area to the difference from the areas of the bounding box and of the ellipse it can be seen that the shape is more related to an ellipse than to a rectangle. For triangles the difference is more obvious since its area is much smaller than the bounding box and a simple threshold is enough to decide if the shape is a triangle or not. This method works very well once the thresholds are found and it is also very fast.

In order to obtain a rotation invariant method, the rotation of the shape should be done until the bounding box area is minimum, but as only one parameter has to be optimized, the result is the global maximum and it is obtained fast without complex optimization methods.

Method 4: Recognition using Least-Squares optimization

The application developed for the 2D recognition of the noisy contours compares the least square errors for the geometrical shapes fitted to the data. The least square error for a geometrical shape with parameter vector \mathbf{p} , e.g. radius, side length, etc., is defined as the sum of minimum distances from the points in the curve to the geometrical shape, e.g.

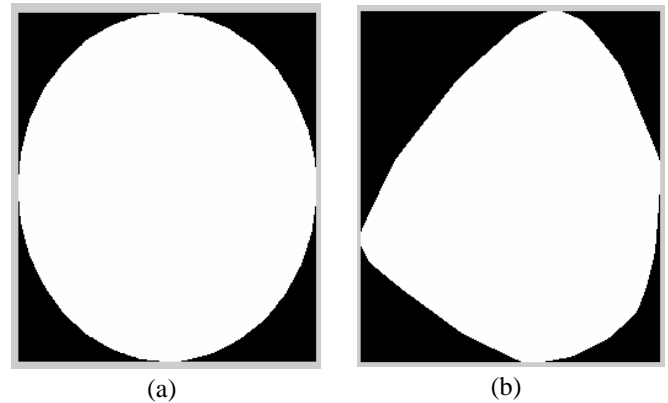


Fig. 14. Bounding box and fitting ellipse on the left, real shape on the right

$$F_p(x) = \frac{1}{2} \sum_{i=1}^m [f_i(x)]^2 \quad (8)$$

where $f_i(x)$ is the minimum distance of point i to the geometrical shape with parameter vector \mathbf{p} .

To find the optimum parameter vector \mathbf{p} giving the minimum total error corresponding to geometrical shapes: circles, rectangles, triangles and ellipses; Levenberg-Marquardt non-linear least squares minimization for unconstrained optimization is used [31], [32]. The geometrical shape giving the minimum sum of least square errors is identified as the best match for the contour drawn by the user. Vector \mathbf{p} corresponding to the geometrical shape with minimum total square error is used to display the recognized shape to the user. The application for the 2D shape recognition consists of three major parts; namely, input of the contour information, functions calculating least square errors for various elementary geometrical shapes and the nonlinear least square algorithm.

The information for the contour drawn by the user is captured by the movements of the hands and is passed as a vector of points to the application. As the provided points are noisy and scattered, a filter is applied before the recognition algorithm is called. The filter resamples the points in the contour uniformly so that the distance between two successive points is kept constant. This prevents the erroneous effect of clustering of the points at the beginning and the end of the drawn contour. The filtered points are then passed to the shape recognition functions.

The second part of the application consists of functions calculating least square errors for elementary shapes such as circles, ellipses, triangles and rectangles. These functions mainly map a parameter vector $p \in R^m$ to an estimated measurement vector $\hat{x} = f(p)$, $\hat{x} \in R^n$ where $n > m$. An initial parameter estimate p_0 is provided from the properties of the data, e.g. mean, max and min of the data points. The measurement vector is simply the distance of a point in the contour to the geometrical shape. The parameter and the measurement vectors are passed to the optimization function for finding the optimum parameters giving the minimum measurement error $\|e\|$ which is defined as:

$$\|e\|^2 = \|x - f(p)\|^2$$

where x is the measurement and $f(p)$ is its estimate.

As for the nonlinear least square method, Levenberg-Marquardt(LM) method is used for finding the parameters of the geometrical shapes. LM method is an iterative technique that finds a local minimum of a multivariate function that is expressed as the sum of squares of nonlinear functions. It can be thought of as a combination of steepest descent and the Gauss-Newton method. When the current solution is far from the correct one, the algorithm behaves like a steepest descent method: slow, but guaranteed to converge. When the current solution is close to the correct solution, it becomes a Gauss-Newton method. LM function calls the functions calculating the errors for principal shapes and returns the optimum parameters. The flow chart of the algorithm can be found in Figure 15.

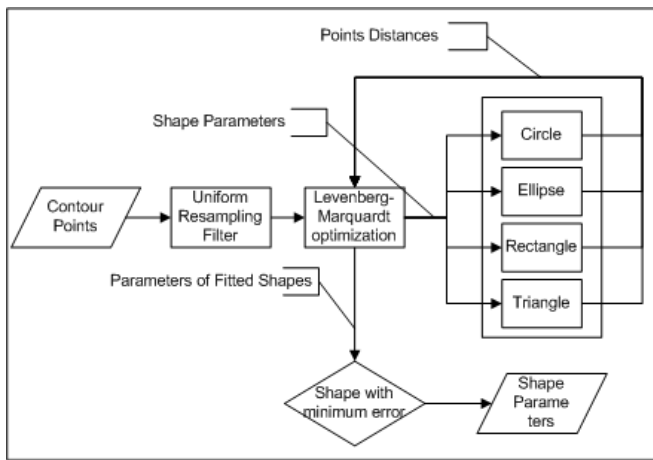


Fig. 15. Algorithm flow chart

The hand controlled mouse modality in this project allows the user to sketch contours in the Virtual Assembly Platform with the aim of creating 3D primitive geometrical shapes such as cylinders, square prisms, rectangular or triangular prisms from 2D sketches. The user draws a contour which is then recognized as a circle, a rectangle or a triangle. The recognized shape is used as a base for creating corresponding 3D shape such as circles for cylinders, rectangles for rectangular prisms etc.. The application for the extraction of 2D primitive geometrical shapes from the contours drawn with the hand controlled mouse modality has to be robust enough; because, the contours drawn by the user may be erroneous, incomplete and noisy; the points may be scattered, concentrated at the beginning or at the end of the contour or there may be unintentional lines drawn before the user stops sketching. Some contours drawn with the hand controlled modality are shown in Figure 16.

To evaluate the performance of the algorithm developed for 2D shape recognition, a database consisting of 100 contours is used where 98 shapes were detected correctly. The database is obtained by letting users to draw contours on the Virtual Assembly Platform using the hand controlled mouse modality.

All four sketch recognition approaches produced recognition rates close to 100%. For the final system method 4 was used due to its robustness, configurability and extensibility. It is not based on relative thresholds and not absolute ones which

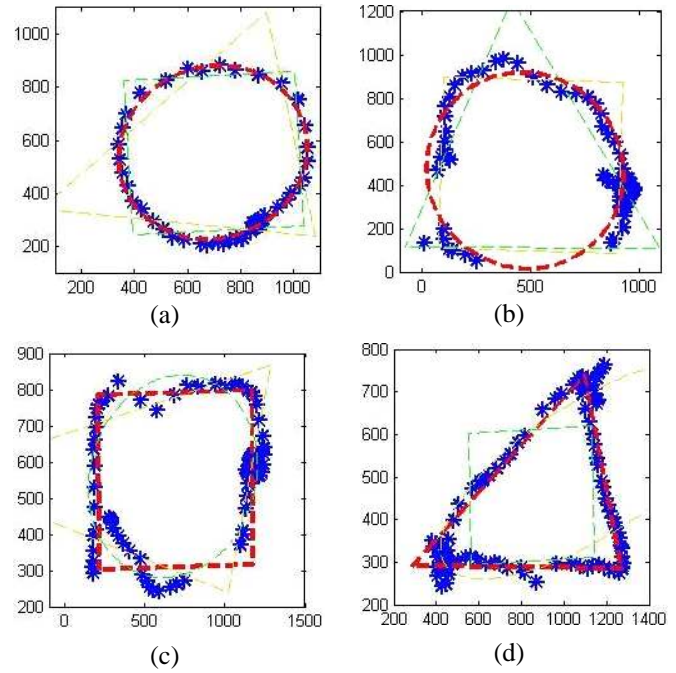


Fig. 16. a) A proper circular contour drawn by the user is shown in blue. The detected shape is drawn in red. b) An incomplete circular contour. c) A noisy rectangular shape d) A triangle with tails at the beginning and at the end

means that the results are more robust. Its only drawback for our case is the computation time. It was sufficient for our application but if real time sketch recognition is needed, the choice should be done on one of the methods 2 or 3 which are fast enough for real time. Moreover mix of different methods could be used.

3) *Deforming the geometries:* The range of objects, which can be built only by assembling primitives, is limited because irregular or non-symmetric shapes can not be drawn. Therefore the proposed 3D query model generation scheme using sketches integrates an object deformation procedure. It is implemented so as to interactively deform the initial 3D-primitive model generate a more complex one. The proposed geometric deformation technique, directly affects the triangulated model of a 3D-object.

Initially, a number of control points are defined on the surface of the object. In order to deform the mesh, the user has to translate its control points. This corresponds to setting a translation vector T_h for each control point. Translation T_h is propagated to the mesh and affects only the closest vertices to the control point. In particular it affects only the points x inside the geodesic window GW , which is defined as follows:

$$GW_u = \{x | \forall x \in V, g(u, x) < \epsilon\} \quad (9)$$

where ϵ defines the window size, u is the control point and $g(u, x)$ the geodesic distance between x and u , i.e. the non-Euclidean distance on the surface.

All vertices lying inside the geodesic window are translated using the following equation:

$$\mathbf{T}(x) = \mathbf{T}_h \cdot K(x, u) \quad (10)$$

where $K(\mathbf{x}, \mathbf{u})$ is a gaussian kernel used so as to assure smooth transition of the translation of the influenced vertices.

$$K(\mathbf{x}, \mathbf{u}) = e^{-\frac{g(\mathbf{x}, \mathbf{u})}{2\delta^2}} \quad (11)$$

where \mathbf{x} is the position of the vertex and δ the standard deviation of the gaussian kernel. An important parameter which strongly affects the result of the deformation is the size ϵ of the geodesic window. Figures 17a and 17b illustrate the results of using two different values for ϵ . Notice that in the second case most of the points are deformed. In the context of the proposed framework the geodesic window is adjusted as the mean distance between two neighboring control points so as to make the user capable of deforming the whole surface of the object.

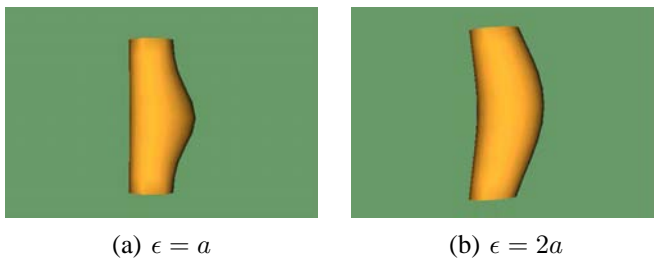


Fig. 17. Deformation using different values of ϵ

It should be also mentioned that in order to produce efficient deformations resampling is performed on the objects' surface so as to make it more regular and to assure smooth deformations.

4) *Building the query-model*: After the primitive objects are imported into the screen and processed accordingly (scaled, rotated, deformed) they are translated to the desired position so as to represent the targeting object. Next, they are grouped into a single object using speech commands. The resulting object is exported and its descriptor is extracted using the method described in Section III-B. Finally, the retrieved objects are imported into the scene, ordered in decreasing similarity, using speech commands.

V. ACTION VERIFICATION MODULE

To have feedback about the machine processing, a talking head was included. It provides audiovisual information about the recognized voice commands and the head and hand tracking.

The talking head used in this project was developed as a cloned 3D appearance with articulation gestures of a real human [33], [34]. The eye gaze and head orientation of the clone can be controlled independently to look at the user or where the user is looking on the screen. The virtual neck is also articulated and can accompany the eye-gaze movements. The audiovisual messages can either be recorded by the original human speaker, or synthesized from text input.

Most model-based and image-based systems describe the influences of movements like lip protrusion or jaw oscillation in limited regions of the face, whereas in reality articulatory movement produces deformations all over the face. For example the nose wings move during speech production and some

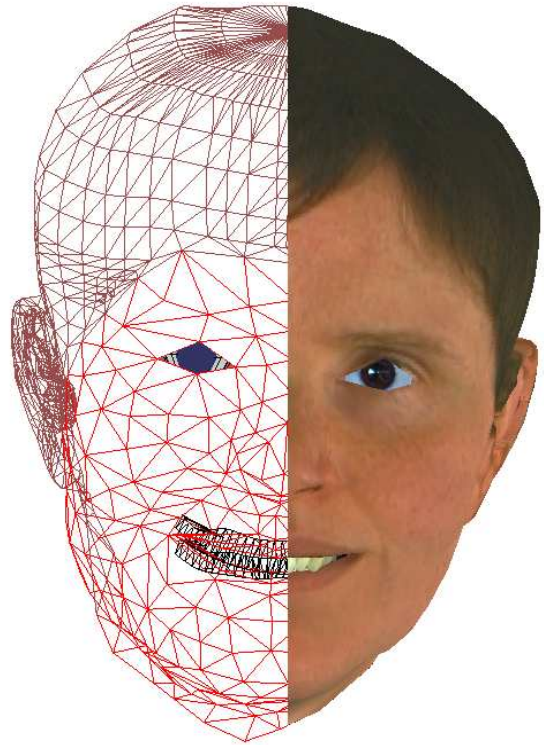


Fig. 18. The talking head

lingual and laryngeal movements have visible consequences on the cheeks and the throat. For the clone, it was an important issue to comprehend the influences by the subtleties of facial deformations induced by the underlying articulatory movements of visible speech. The facial animation is based on the recording of a French speaker, producing 40 prototypical configurations. His face was furnished with 166 colored beads. In a statistical analysis of the 3D points, six parameters could be identified that sufficiently model the facial movements of the lower part of the face.

For the multimodal interface the clone is used as a feedback of computer activity [35]. It aims to provide a more natural interface for the control of sketching and processing of shapes, that uses strategies humans use in face-to-face interaction. The aim of the clone as it is used in this application is to provide an appropriate feedback by the computer imitating the strategies a human might use. The clone greets the user when he or she is detected for the first time by the motion capture system. When the user is lost or localized again by the system, the clone makes corresponding utterances. During the sketching, when a hand is found and tracked, it follows the virtual pen drawing on the screen. Once the hand is lost or no hand has been localized it looks at the user's face. This provides information that a user can understand intuitively.

To confirm the recognition of voice commands, the clone announces the actions of the system that will follow. As some voice commands are used very often, such as for example "select", a random choice between several messages is provided, to avoid annoying repetitions of utterances by the

clone. In the current implementation, only French audiovisual speech synthesis for the clone was available.

VI. APPLICATION DEMOS

The proposed framework was tested in two scenarios, the assembly of a piston and the 3D content based search using as query a model generated using sketches and primitive objects.

A. Piston Assembly

In this scenario, the user had to assemble a piston using the developed gesture-speech interface. Specific speech commands described in Section IV-A.A were used to select an action and then the objects are manipulated using gestures. Figures 19a-d illustrate four consecutive steps of the procedure.

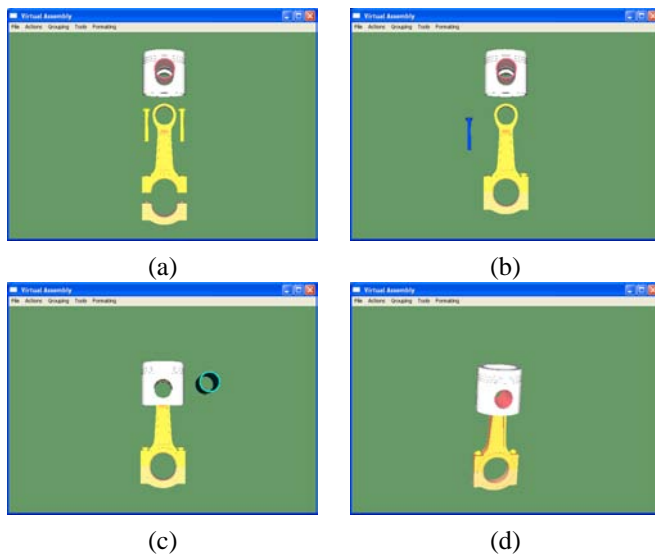


Fig. 19. Gesture-Speech based assembly of a piston

B. 3D content-based search using sketches

In this scenario the user has to draw primitive objects using sketches to assemble them into a more complex one and to search into the database for similar objects using the 3D content-based search engine (Figure 20).

Figure 21 illustrates four snapshots of the procedure of trying to design a car and to search for similar content. Notice that from the retrieved objects only the 8th is not a car as illustrated in Figure 21d.

Figure 22 illustrates four snapshots of the procedure of trying to design a car and to search for similar content. Notice that all the retrieved objects are chairs 22d.

VII. CONCLUSIONS

In the present report we described the methods and results of Project 7 of the eINTERFACE-2005 summer workshop for multimodal interaction. The results of this work illustrate the efficiency of using multimodal interfaces to guide multimedia and VR applications. In particular, the gesture-speech controlled virtual assembly application has been observed to

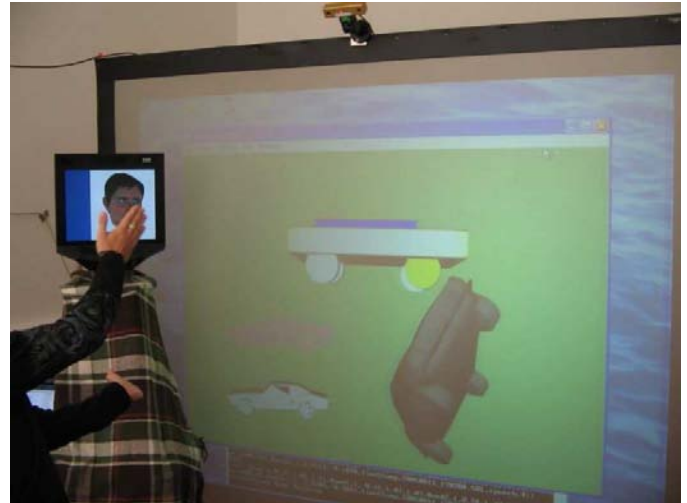


Fig. 20. Using the MASTER-PIECE platform - rotating a car

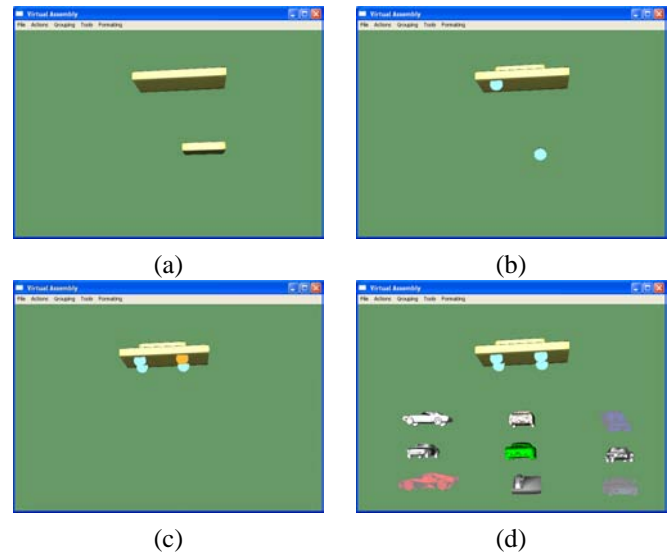


Fig. 21. Car sketching and recognition

increase the immersion of the user to the application due to the physical interaction of the user with the system. Finally, the presented sketch-based query model generation for 3D search eliminates the demand of an existing query model in order to perform 3D search and provides the user with a natural means of generating in 3D the query object.

VIII. ACKNOWLEDGEMENTS

This work was supported by the EU funded SIMILAR Network of Excellence.

REFERENCES

- [1] "W3c workshop on multimodal interaction," 19/20 July, 2004, Sophia Antipolis, France (<http://www.w3.org/2004/02/mmi-workshop-cfp.html>).
- [2] "Special issue: Interacting with emerging technologies, j. strickon, guest ed., iee computer graphics," Jan-Feb 2004.
- [3] I. Marsic, A. Medl, and J. Flanagan, "Natural communication with information systems," in *Proc. of the IEEE*, 88, 8, Aug. 2000, pp. 1354–1366.

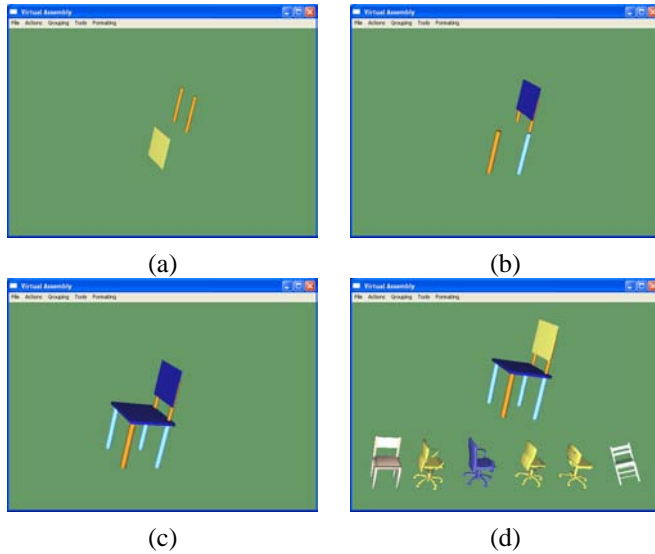


Fig. 22. Chair sketching and recognition

- [4] J. Lumsden and S. A. Brewster, "A paradigm shift: Alternative interaction techniques for use with mobile & wearable devices," in *Proc. 13th Annual IBM Centers for Advanced Studies Conference CASCON'2003, Toronto, Canada*, 2003, pp. 97–110.
- [5] T. V. Raman, "Multimodal interaction design principles for multimodal interaction," in *CHI 2003, Fort Lauderdale, USA*, 2003, pp. 5–10.
- [6] A. Camurri and G. Volpe (Eds.), "Gesture-based communication in human-computer interaction," in *Lecture Notes in Artificial Intelligence*, no. 2915, Springer Verlag, February, 2004.
- [7] S. Berchtold and H.P. Kriegel, "S3: Similarity search in cad database systems," in *Proc. of SIGMOD*, J. Peckham, Ed. ACM, 1997, pp. 564–567.
- [8] H.P. Kriegel M. Ankerst, G. Kastenmuller and T. Seidl, "3d shape histograms for similarity search and classification in spatial databases," in *Proc. of the 6th Int. Symposium on Spatial Databases (SSD1999)*, Hong Kong, China, 1999.
- [9] U. Schurmans, A. Razdan, A. Simon, P. McCartney, M. Marzke, D. V. Alfen, G. Jones, J. Rowe, G. Farin, D. Collins, M. Zhu, D. Liu, and M. Bae, "Advances in geometric modeling and feature extraction on pots, rocks and bones for representation and query via the internet," in *Computer Applications in Archaeology (CAA)*, 2001.
- [10] E. Paquet and M. Rioux, "A content based search engine for vrml databases," in *Proc. of IEEE Conf. On Computer Vision and Pattern Recognition (CVPR1998)*, 1998.
- [11] E. Paquet and M. Rioux, "Nefertiti: A tool for 3-d shape databases management," *SAE Transactions: Journal of Aerospace*, vol. 108, pp. 387–393, 2000.
- [12] M. T. Suzuki, "A web-based retrieval system for 3d polygonal models," in *Joint 9th IFSA World Congress and 20th NAFIPS Int. Conf. (IFSA/NAFIPS2001)*, Vancouver, 2001, pp. 2271–2276.
- [13] J. Löffler, "Content-based retrieval of 3d models in distributed web databases by visual shape information," in *Proc. of Int. Conf. on Information Visualisation (IV2000)*, 2000.
- [14] C.M. Cyr and B. Kimia, "3d object recognition using shape similarity-based aspect graph," in *Proc. of Int. Conf. on Computer Vision (ICCV2001)*, 2001, pp. 254–261.
- [15] D. Zhang and M. Hebert, "Harmonic maps and their applications in surface matching," in *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition (CVPR99)*, 1999.
- [16] G. Mori, S. Belongie, and J. Malik, "Shape contexts enable efficient retrieval of similar shapes," in *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition (CVPR2001)*, 2001.
- [17] A. E. Johnson and M. Hebert, "Using spin-images for efficient multiple model recognition in cluttered 3-d scenes," *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI)*, vol. 21, no. 5, pp. 433–449, 1999.
- [18] G. Blais and M. Levine, "Registering multiview range data to create 3d computer objects," *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI)*, vol. 17, no. 8, pp. 820–824, 1995.
- [19] C. S. Chua and R. Jarvis, "3d free-form surface registration and object recognition," in *Proc. of Int. Journal of Computer Vision*, Kluwer Academic Publishers.
- [20] B. J. Besl and N. D. McKay, "A method for registration of 3-d shapes," *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI)*, vol. 14, no. 2, pp. 239–256, 1992.
- [21] H. Hugli, C. Schutz, and D. Semitekos, "Geometric matching for free-form 3d object recognition," in *ACCV, Singapore*, 1995, pp. 819–823.
- [22] H. Hugli and C. Schutz, "Geometric matching of 3d objects: Assessing the range of successful configurations," in *Proc. of Int. Conf. of Recent Advances in 3-D Digital Imaging and Modeling*, Ottawa, Ontario, Canada, May 1997.
- [23] D.Tzovaras P.Daras, D.Zarpalas and M.G.Strintzis, "3d model search and retrieval based on the spherical trace transform," in *IEEE Intl Workshop on Multimedia Signal Processing*, Sienna, Italy, 2004.
- [24] D.V. Vranic and D. Saupe, "Description of 3d-shape using a complex function on the sphere," in *Proceedings IEEE International Conference on Multimedia and Expo*, 2002, pp. 177–180.
- [25] K. Nickel, E. Seemann, and R. Stiefelhagen, "3d-tracking of head and hands for pointing gesture recognition in a human-robot interaction scenario," in *IEEE International Conference on Automatic Face and Gesture Recognition*, Seoul, Korea, 2004, pp. 565–570.
- [26] N. Jovic, B. Brumitt, B. Meyers, and S. Harris, "Detecting and estimating pointing gestures in dense disparity maps," in *IEEE International Conference on Face and Gesture recognition*, Grenoble, France, 2000, pp. 468–475.
- [27] S. Carhini, J. E. Viallet, and O. Bernier, "Suivi statistique simultane des parties du corps pour des interactions bi-manuelles," in *ORASIS*, Fournol, France, 2005.
- [28] R. Feraud, O. Bernier, J.E. Viallet, and M. Collobert, "A fast and accurate face detector based on neural networks," *Pattern Analysis and Machine Intelligence*, vol. 23, no. 1, pp. 42–53, January 2001.
- [29] K. Bartkova and D. Juvet, "Modelization of allophones in a speech recognition system," in *ICPhS (International Congress of Phonetic Science)*, Aix-en-Provence, France, 1991, pp. 474–477.
- [30] R. Schwartz and Y.L. Chow, "The n-best algorithm: an efficient and exact procedure for finding the n most likely sentence hypothesis," in *ICASSP (International Conference on Acoustic Speech and Signal Processing)*, Albuquerque, USA, 1990, pp. 81–84.
- [31] K. Levenberg, "A method for the solution of certain non-linear problems in least squares," *Quarterly of Applied Mathematics*, vol. 2, no. 2, pp. 164–168, 1944.
- [32] D.W. Marquardt, "An algorithm for the least-squares estimation of nonlinear parameters," *SIAM Journal of Applied Mathematics*, vol. 11, no. 2, pp. 431–441, 1963.
- [33] F. Elisei G. Bailly, M. Berar and M. Odisio, "Audiovisual speech synthesis," *International Journal of Speech Technology*, , no. 6, pp. 331–346, 2003.
- [34] G. Bailly L. Reveret and P. Badin, "Mother: a new generation of talking heads providing a flexible articulatory control for video-realistic speech animation," in *International Conference on Speech and Language Processing*, 2000, pp. 755–758.
- [35] F. Elisei S. Raidt and G. Bailly, "Face-to-face interaction with a conversational agent: eye-gaze and deixis," in *International Conference on Autonomous Agents and Multiagent Systems*, Utrecht, Netherlands, 2005.
- [36] PTGREY, "http://www.ptgrey.com," retrieved in 2005.

IX. APPENDIX: SOFTWARE NOTES

A. How to use: Assembly application

The virtual assembly application supports both mouse-keyboard and the multimodal gesture-speech interface.

For performing virtual assembly the user has to:

- Select one of the available assembly scenarios in "xml" format available in the "scenes" folder.
- From the "Action Menu" select "Start Assembly"
- Manipulate the objects, either using mouse-keyboard and the actions in the "Action Menu" or using gesture and speech as described in the following subsections
- When two objects that should be connected are close together they will snap.

For performing sketch-based 3D query model generation the user has to:

- Import the required primitive objects in the scene using either from the menu "File->Insert Primitive" or using sketches by "Actions->Start Sketching" or using gesture and speech as described in the following sections
- Manipulate the objects, either using mouse-keyboard and the actions in the "Action Menu" or using gesture and speech as described in the following subsections
- Group primitives into a single object: Use "Group->Select Group" to initiate grouping. Click on each sub-object and to "Group->Select Next" until all sub-objects are selected. Finally click on "Group->Finalize Selection". Alternatively, the corresponding gesture-speech commands can be used as described in the following subsections.
- Finally, use "File->Save Selected" to save the object and use "Actions->Search for Similar" to search for similar content.
- Only after the 3D descriptor extraction is terminated use "Actions->Retrieve Object" to retrieve the similar 3D objects from the database.

B. How to use: 3D search engine

The 3D search engine uses as input a 3D triangulated model in VRML format. The procedure for performing 3D search is demonstrated in the batch file preprocess.bat. In particular the user has to follow the next steps:

- 1) Generate the voxel model: Execute "*R3DST_A* name.WRL", where "name.WRL" is the name of the VRML file.
- 2) Generate initial descriptors: "*R3DST_B* name.vm K 16 3", where "name.vm" the name of the voxel model file, and "K", "16", "3" parameters of the algorithm.
- 3) Generate final descriptors: "*R3DST_C* name.Kraw_00 25", where "name.Kraw_00" are the initial descriptors
- 4) Finally, file "*ST_Descr.Kraw_00.txt*" is generated, which contains the final descriptors.
- 5) Perform matching: "*Retrieve.exe ST_Descr.Kraw_00.txt DatabaseName*", where DatabaseName is either ITI, PB or All corresponding in searching in the ITI the Princeton or both databases.
- 6) The retrieved object are listed in the resultsObjectName.txt file where ObjectName is the name of the query model.

C. How to use: Gesture recognition

1) *Presentation*: The gesture program system needs a firewire stereo camera Bumblebee from Pointgrey [36]. It works under linux (Mandrake Linux 9.2 3.3.1-2mdk) and has been compiled with gcc version 3.3.1 on a Pentium IV 3Ghz computer.

The executable file <pointage> takes the images from camera, extract user body parts position (head and hands). Then, the direction pointed by the user and the angles between the hands are computed. These data can be sent through a pipe to another application or through the network using socket with the following format:

```
Head,Hb,Hx,Hy,HZ,Hand1,h1b,h1x,h1y,h1z,
Hand2,h2b,h2x,h2y,h2z,Pointer,px,py,
Angles,ab,a0,a1,TimeStamp,YYYYMMDD_HHMMSS.ms
```

With:

—— Head ——

- Hb: head present boolean (true if the head is tracked).
- Hx,Hy,HZ: 3D position of the head in meters from the camera.

—— Hand1 ——

- h1b: first hand present boolean (true if the first hand is tracked).
- h1x,h1y,h1z: 3D position of the first hand in meters from the camera.

—— Hand2 ——

- h2b: second hand present boolean (true if the second hand is tracked).
- h2x,h2y,h2z: 3D position of the second hand in meters from the camera.

—— Pointer ——

- px,py: the location pointed by the user on the screen in pixels.

—— Angles ——

- ab: valid angles boolean (true if the two hands are tracked and if at least one of the hand is in front of user head)
- a0,a1: angles α and β of the hand.

—— TimeStamp ——

- YYYY: year MM:month DD:day HH: hour MM:minute SS:second ms: millisecond

For example when nobody is present the output of the system is:

```
Head,0,0.00,0.00,0.00,Hand1,0,0.00,0.00,0.00,
Hand2,0,0.00,0.00,0.00,Pointer,-1280,-1024,
Angles,0,0.0000,0.0000,TimeStamp,20050809_171000.106
```

2) *Configuration and calibration*: To configure the gesture application, the value of several parameters can be set in <rep_config/pointage.cfg>.

To configure the output socket or pipe:

- FLUX_SORTIE: Set to 1,2,3 to use a pipe to give data to another application on the same machine. Set -1 to use a socket with a TCP/IP connection to give data to a computer on the network.
- IPX_ADRESSE: four number for the ip adress of the remote computer when *FLUX_SORTIE* = -1.

- PORT: port used when $FLUX_SORTIE = -1$.

To calibrate the camera and the screen (figure 23), some parameters depending on their physical positions need to be set (all distances are in meters):

- DISTANCE_X_CAMERA_CENTREIMAGE: distance between the middle of the image and the camera on X axis.
- DISTANCE_Y_CAMERA_HAUTIMAGE: distance between the upper bound of the image and the camera on Y axis.
- DISTANCE_Z_CAMERA_IMAGE: distance between the screen and the camera on Z axis.
- ANGLE_CAMERA: angle between camera axis and Z axis.
- HAUTEUR_CAMERA: distance between the camera and the ground on Y axis.
- TAILLE_IMAGE_X, TAILLE_IMAGE_Y: horizontal and vertical size of the image.

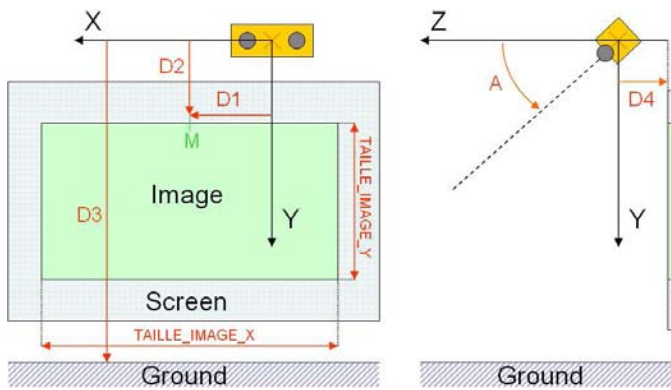


Fig. 23. Parameters for the camera and image: (D1) DISTANCE_X_CAMERA_CENTREIMAGE. (D2) DISTANCE_Y_CAMERA_HAUTIMAGE. (D3) HAUTEUR_CAMERA. (D4) DISTANCE_Z_CAMERA_IMAGE. (A) ANGLE_CAMERA.

Some parameters defines the limits of the interaction field (in meters):

- XE_MIN, XE_MAX: min and max X value for 3D coordinates.
- YE_MIN, YE_MAX: min and max Y value for 3D coordinates.
- ZE_MIN, ZE_MAX: min and max Z value for 3D coordinates.

The camera must be calibrated correctly.

- RESOLUTION_CAMERA_X, RESOLUTION_CAMERA_Y: work resolution of the camera (in pixels).
- B_EXPOSURE: exposure parameter for the camera.
- B_RED, B_BLUE: white balance setting (from 0 to 64).

The value of the parameters of the white balance depends on lighting condition. To find the good value for these two parameters (B_RED, B_BLUE) an automatic white balance procedure should be performed. A large white paper must be shown in front of the camera and $< W >$ key pressed on the keyboard while gesture application is running. The

paper must represent the main part of the centre of the image. Once $< W >$ is pressed, the paper must be hold during about 10s. Then, in the console, the value for red and blue parameters are display in one of the last line (for example: $< \text{---Valeursactuellesdelacamra : R : 13B : 47Exp : 400} >$ means that B_RED=13 and B_BLUE=47 are the correct value). The other parameters of the config file concern the detection and tracking of body parts and should not be changed.

D. How to use: speech recognition

The speech recognition use the executable files:

$< \text{../paroleV4/iomshell} >$,

$< \text{../paroleV4/bigfif} >$

and $< \text{../paroleV4/tst_mkvnb} >$

with the model:

$< \text{../modeles/echec_20050107_F15x27c_g08_R0p.gkz} >$

for the vocabulary definition. To test the speech recognition only, run the script $< reco >$. To change the number of n-best results given change the parameter $< -sol = >$. The first lines output when a word is recognize are:

20050811,00711.858 speech_start

20050811,00712.083 speech_end

!EXP: Src=si.inl;Type=f10.r10;File=5;

Then each n-best result is given on a separate line:

Sol=1 ; Dec="balou" ; Score=1155 ; Gscore=-60976 ;

NbFrames=38 ; Bnodes=81 ; Time=50 ;

Where $< Sol = >$ is the rank of the n-best solution, $< Dec = >$ contain the word recognized and $< Score = >$ is the solution score.

The speech commands available are :

TABLE I
SPEECH COMMANDS

Nb	Speech Command	Action
0	no speech	no action
1	mougli	move
2	lâche	stop action
3	sélectionne	select object
4	balou	rotate object
5	chercanne	scale object
6	éché	search object
7	click	select group
8	prend la reine	select next object
9	tour prend	end select group
10	cavalier prend	retrieve an object
11	pose	save object
12	met le roi	delete object
13	met la reine	clone object
14	O.K. là	start sketching

E. How to use: fusion

The file $< fusion >$ mix the speech recognition and the gesture recognition and send all to the 3D application via a socket connection. When gesture is mixed with speech, the parameter FLUX_SORTIE must be egal to 3 so that gesture application output results on a pipe to fusion application.

To change the IP adress of the computer where the 3D application is, change the value of ADRESSE_IP1 in the function void Acquerir().