A Lightweight Speech Detection System for Perceptive Environments

Dominique Vaufreydaz, Rémi Emonet, Patrick Reignier

A Lightweight Speech Detection System for Perceptive Environments

Dominique Vaufreydaz, Rémi Emonet, Patrick Reignier

PRIMA - INRIA Rhône-Alpes, Zirst, 655 avenue de d'Europe,
Montbonnot, 38334 Saint Ismier cedex, France
{Dominique.Vaufreydaz,
Remi.Emonet,Patrick.Reignier}@inrialpes.fr
http://www-prima.inrialpes.fr/

Abstract. In this paper, we address the problem of speech activity detection in multimodal perceptive environments. Such space may contain many different microphones (lapel, distant or table top). Thus, we need a generic speech activity detector in order to cope with different speech conditions (from close-talking to noisy distant speech). Moreover, as the number of microphones in the room can be high, we also need a very light system. The speech activity detector presented in this article works efficiently on dozens of microphones in parallel. We will see that even if its absolute score of the evaluation is not perfect (30% and 40% of error rate respectively on the two tasks), its accuracy is good enough in the context we are using it.

1 Introduction

The base principle of research in ubiquitous computing is to make the computer disappear from the human computer interfaces. Classical input and output devices (keyboard, mouse, screen, etc.) are replaced by other, less intrusive modalities such as voice recognition or computer vision. In order to conduct research in this domain, many research laboratories have equipped dedicated rooms with multiple sensors (cameras, microphones, etc.) and perceptual software (2D and 3D visual tracking systems, speech recognition systems, etc.). The goal is to enable the computer system to understand what the user is saying or doing. The computer system is then able to behave in accordance with the user's intentions. Such highly equipped spaces are often called perceptive environments.

In these environments, all audio sensors, from simplest ones to most complicated ones, have an important role to play. Speech is indeed one of the preferred and most natural communication channels in human to human interactions, and sounds are revealing of human activity. This is why many perceptual environments, such as in the CHIL project [1], are equipped with speech detection, speech recognition and acoustic localization systems. One requirement in such perceptive environments is to be able to process multiple and various microphones in parallel while fitting real time constraints.

Within the CHIL project [1], we are developing a speech detection system that fulfills the requirements of these perceptual environments. Although much research has already been conducted on this point and different approaches have been proposed (such as [2] and [3]), the problem is still open. In addition, we impose two constraints to what is done in most of other systems. Our system must be autonomous. It must be started and then run without human action. It must require neither training nor tuning each time the operating conditions change. We also want to keep our system as light as possible.

In section 2, we first give a description of our speech detection system. Section 3 then presents evaluations that were conducted and the results obtained in the NIST 06s evaluation[1]. Finally we will give a conclusion and some further works to be carried out in order to improve our system.

2 System description

In our perceptive environment, the full SAD (Speech Activity Detection) system can run at the same time over one or many microphones. In this last case, there are two kinds of answer. First, a SAD decision is made at least for each microphone. We can also define a set of microphones in order to get an “ambient" SAD decision using for example multiple microphone arrays. A majority vote is done among all microphones to determine the current state. If speech and non speech votes are equal, the state of the global answer remains the same. This strategy is not optimal when using a large set of different microphones. The design of the system was made in order to run several systems in parallel over multiple groups of microphones.

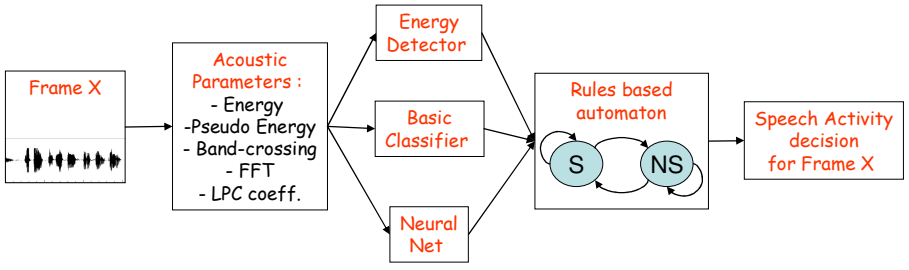

Fig. 1. Design of the SAD System.

On each input, the current version of our SAD system works using several sub-systems: an energy detector, a basic classifier and a neural net trained to recognize voiced segments like vowels for example. At each timestep, i.e. for each frame, each sub-system gives a speech or non-speech answer. Then a hand-made rule-based automaton determines the final result: whether or not there is speech activity. This tools is designed to be enhanced with complementary other subsystems.

[1] See http://www.nist.gov/speech/tests/rt/rt2006/spring/.

2.1. Energy detector

The energy detector uses pseudo energy (we do not sum square values of samples but only absolute values) to determine variation of the input signal energy. It works using two couples of time delay/energy threshold: *TimeOn/EnergyOn* and *TimeOff/EnergyOff*. Simply speaking, if the pseudo energy goes over *EnergyOn* during *TimeOn*, the energy detector emits a *START_SPEAKING* event as a result. In the same way, if the pseudo energy falls under *EnergyOff* during *TimeOff*, the result is *STOP_SPEAKING*. For stable periods, the return values are respectively *STILL_SPEAK* and *NOT_SPEAK*.

As the system is designed to run permanently, it is obvious that these energy thresholds cannot always reflect the current voice/background noise energies. We added to the energy detector the ability to adapt dynamically these thresholds. A sliding window of 50 seconds of previous values for speech portion energy is maintained. In order to smooth threshold changes, the window is filled with the *EnergyOn* value at the initialization time. Then, when the global SAD state changes, for example when final answer of the SAD system goes from speech to non-speech, the training system computes the new threshold value. The system does exactly the same computation on a separate sliding window for *EnergyOff*.

In order to prevent usage of outliers in the online adaptation process, we do not use data from the beginning and the ending of speech or non-speech segments. We privilege inside segments which should be more stable. Thus, we eliminate *TimeOn* data from the beginning and *TimeOff* at the end of speech segments for our online adaptation. Identically, we do not use *TimeOff* and *TimeOn* data respectively from beginning and ending of non-speech segments. We also do not use data from segments that are not long enough. So, if a segment does not contain at least 1 second of interesting data, it is not use to compute new thresholds.

The final computation of new values for *TimeOn* and *TimeOff* do not use a simple average approach but a “median” one. We sort the adaptation data contained in the sliding windows and remove possible outliers, i.e. too low and too high values. At least, we keep only 60% of the data in order to re-estimate our thresholds. Our real adaptation time is thus 30 seconds (60% of 50s) for each threshold.

2.2. Basic Classifier

This classifier is dedicated to recognize and to tag specific sound classes: fricatives, low frequency sounds like computer or air conditioning fans, and other sounds. The first step is a hamming window and a Fast Fourier Transform (FFT) [4] to obtain the spectrum of the signal. The classifier deals with 5 identical sub-frequency bands from 1 to 8000 hertz where it computes energy. This classifier works only on signal recorded at 16000 hertz or higher sampling rates. In this last case, frequencies over 8000 hertz are not used.

With the 5 energy values, the module can classify the audio signal:

- if more than 90% of the total energy is concentrated in the 2 lowest bands, the sound is a *low frequency sound*
- if the energy in the 2 lowest bands is less than in all other bands, the sound is a *fricative*
- in all other cases, the sound remains *unclassified*.

2.3. Neural Network

The neural net is a multi-layer perceptron with 2 hidden layers. It uses as input coefficients computed on the input frames:

- Zero crossing: number of time the signal goes from a negative to a positive value and vice versa. Actually, we use a variant called band-crossing [5] that does not count oscillations in a band around 0.
- Energy: the sum of the square values of samples.
- 16 predictor coefficients: they are extracted from a speech analysis method called Linear Predictive Coding (LPC) [6]. We use the auto-correlation method combined with the Durbin recursion to compute them.

This module is the only sub-system that needs to be trained. The training was made once and for all on 1 hour of French speech extracted from the BREF corpus [7]. The phonetics labels used during the training phase are not the original BREF ones but were computed with RAPHAEL [8], a French recognizer. The training data were almost equilibrated, ~50% of female voice and ~50% of male voice. Result of this module can be *speech* or *non speech*.

2.4. Rules based automaton

This automaton is designed to integrate results from all subsystems to produce a final answer. It consists in 2 states (*speech* and *non speech*) with hand-made rules to change from one state to the other. The rules were defined using knowledge about each subsystem. In all cases, if all subsystems agree on the current state, i.e. when each result is a speech one[2], the system uses it. In the *speech* state, if the energy detector return value and at least one another result are *non speech*, we go into the *non speech* state. We do the same when the basic classifier returns *unclassified* and the neural net *non speech* or when the basic classifier gives *low frequency sound* as answer. Symmetrically, we defined the same type of rules for the *non speech* state.

[2] Speech events are *START_SPEAKING* or *STILL_SPEAK* for the energy detector, *fricative* for the basic classifier and *speech* for the neural net.

3 Evaluation

3.1. Implementation

The implementation of the full system is made in C++ and can run on multiple operating systems. As explained above, the system can handle signal from 16 KHz to 44.1 KHz. During evaluation, the SAD system works on frames of 256 samples on a 16 KHz signal. Thus, the time precision of our system is 16 ms. Another important point: the system remains the same for all evaluation tasks. We do not have specific configuration or training for close talking or far field microphones.

3.2. RT06S evaluation Data

The RT-06S evaluation is focused on the Meeting Domain interaction with two sub-domains (or tasks). The first one consists of ten meetings recorded in a conference room in six different sites: it is called "*confmtg*". The second one, aka "*lectmtg*", is composed of several lectures with lecturer and question/answer speech. Each sub-domain has different sensor setups, different levels of interactions and multiple structures of test excerpts. The reader may refer to [9] to find more detailed information about the evaluation data.

For each task, one may run its speech activity detection system in many different conditions. According to our system characteristics, we decided to evaluate our system on the following subset of conditions:

- Individual head microphone (*ihm*)
- Multiple Distant Microphones (*mdm*)
- Single Distant Microphone (*sdm*)
- All Distant Microphones (*adm*)
- Multiple Source Localization microphone Arrays (*msla*)

The final evaluation contains about 180 minutes of speech for the *confmtg* task and 145 minutes for *lectmtg*.

3.3. Evaluation metrics

In this paper, we use two different metrics in order to test if our system fulfils our needs: a light and accurate system. To check if our SAD system is light enough, we compute the real-time factor as expressed in equation (1).

$$\text{Real - time factor} = \frac{\text{Total processing time}}{\text{Data time}} \quad (\mathbf{1})$$

This factor permits to know easily if the system can be real-time. If the real-time factor is less or equal to 1, so if the processing time is less or equal to the data time,

the system can be considered as real-time. In the next section, we will check how many different inputs we can handle at the same time.

The other evaluation point concerns accuracy. We need to know how efficient our SAD system is. Within the RT06S evaluation, the SAD metric is time based [9]. The scoring system first computes the full speech time over the considered audio signal. Then using output from systems and reference files, manually annotated, we obtain the missed speech and the false alarms times. Doing that on all files from a condition for a given task and accumulating values, we can calculate the overall error rate on the given task using the equation (2):

$$\text{SAD Error} = \frac{\text{Missed speech} + \text{False Alarm time}}{\text{Speech time}} \quad (\mathbf{2})$$

In order to enrich this result, we built tools to compute some extra information. For each task and for each condition, we first extract for each talk, the SAD error rate. Then we decompose the Missed speech time in four time-weighted categories:

- *Full miss*: a complete speech event which is not detected;
- *Miss begin*: the beginning of a speech event is not detected;
- *Miss In*: middle part(s) of a speech event not tagged as speech;
- *Miss end*: the end of a speech event is not found.

Using this new information, we will be able to analyze more precisely the error committed by our SAD system.

3.4. Results

In this section, we detail the results of our system in the RT06s evaluation. First, we will introduce speed measurement and then, the accuracy of our SAD system.

3.4.1 Speed evaluation

As we have already said, speed is a strong constraint for us. We work in interactive spaces and we need low latency application. It is not suitable for us to transmit a lot of data over the network. Thus, we need to be able to process a maximum of microphones on a single computer.

If we measure speed factor on a single 16 KHz audio signal, the computational time for 1 second of speech is 0.0076 second: in theory, the system can handle more than 130 channels at the same time. In practice, operating system scheduling, memory management and multi-channels SAD fusion can alter this result.

We built a test set using 300 seconds segments (containing voice) extracted from the RT06s evaluation database. The full SAD system ran over this set using firstly 1 segment, then 2, etc. Each time, segments were chosen randomly. We stop the test when the real-time constraint was violated, i.e. when the real-time factor goes over 1. The next figure shows the experimental results.

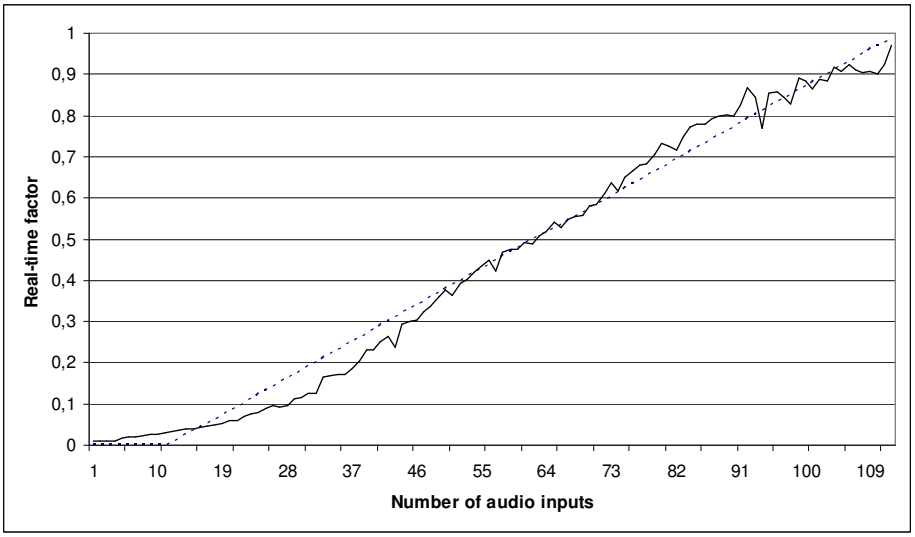

Fig. 2. Speed performance of our SAD system running on a single core unit of a processor and processing multiple inputs.

On this figure, one can found a dashed line obtained by linear regression on data. We also see a curve showing the real-time factor of our system regarding inputs.

We can first see that the real-time factor curve is not always increasing. This surprising phenomenon can be explained by the load of the computer and by the computation time that changes from one file to another (see 2.1). The other result which affords is that our system is linear-like over 50 inputs. It is very important because if we want to add some microphones, we can do it without rethinking the whole computer configuration.

Finally, our SAD system can process 112 streams which is a good score. In real conditions, i.e. when we do not process files, we must also consider that acquisition process will alter this result and certainly slow the system. Nevertheless, we can objectively say that we can process as many microphones as a sound card can record in real time.

3.4.2 Speech Activity Detection accuracy

This section presents experimental results from the RT06s evaluation. In these experiments, starting energy thresholds of the energy detector were values empirically defined during previous research projects (NESPOLE! [10] and FAME [11]). The evaluation metrics of our system are given in the three following tables.

Table 1. Global results over the different tasks.

Task	Condition	Overall Error Rate	Best Error Rate	Worst Error Rate
confmtg	ihm	78,54%	20,31%	1917,99%
	mdm	46,98%	13,12%	80,20%
	sdm	41,26%	18,48%	76,98%
lectmtg	adm	27,81%	3,73%	95,51%
	mdm	32,59%	4,29%	95,47%
	msla	30,61%	3,86%	100,00%
	sdm	33,87%	5,74%	85,29%

The first table above gives us general information about accuracy of our SAD system. The official results of the RT06s evaluation are given in the "Overall Error Rate" column. We can first see that our system is not accurate on the *confmtg-ihm* condition. For this condition, only speech coming from the main speaker must be tagged as speech. As our system is not design to do that (every speech segment can be tagged as speech, even if its energy is low), this result is not really significant: our worst score within this condition is 1918% of error. As we did not understand correctly this task before evaluating, we let the result in the previous table but we will not analyze more precisely this condition. Concerning others conditions of the *confmtg* task, we can see that our average error rate is ~43%. Our best scores, over one seminar, are not good (13% and 18%). We will check in the next section where our system fails. If we look to the *lectmtg* task, we can see that our global results are better: ~30% of error in average over all condition. Moreover, our best scores are good (from 3% to 6%) but our worst score stay high (up to 100%).

We will now trying to understand more precisely where the errors are. The following tables show our additional metrics computed first on the confmtg task, next on *lectmtg*.

Table 2. Detailed results of the *confmtg* task.

Condition	Full Miss	Miss Begin	Miss End	Miss In	False Alarm	Error Rate
mdm	13,22%	3,31%	5,38%	24,97%	0,11%	46,98%
	(28,13%)	(7,04%)	(11,45%)	(53,15%)	(0,23%)	(100,00%)
sdm	1,63%	2,17%	4,39%	26,86%	6,22%	41,26%
	(3,95%)	(5,25%)	(10,63%)	(65,11%)	(15,07%)	(100,00%)

In this table, one can found the official evaluation error rate in the last column. The second sub-row of each entry gives the percentage of each type of error on the overall error rate.

On the multiple distant microphone (*mdm*) condition, our system is not good at all. 13% of the speech segments were entirely missed. 25% of the missed speech is within a speech turn. A good result is the false alarm error rate which is very low. This value is correlated to the 13% of full miss. Our SAD system did not make mistake on non speech segments but was insensible to some speech parts. Our starting thresholds were too high and not adapted to this condition. Moreover the system do not managed to adapt them online. Concerning the single distant microphone (sdm) condition, results are slightly better. In fact, we only have a *full miss* rate of ~2%. We see a rise of the *miss end* and *false alarm* rates. *Miss in* factor stay over 60% of our errors.

If we look globally at the previous table, we can say that ~60% of our errors are due to intra speech undetected portions. Even if we could artificially solve this problem by changing the *TimeOff* delay of our system, we do not want to do it. Field experiments have already shown at our laboratory that adapting system to an evaluation can lead to decrease drastically real world performances. Other metrics can not be averaged because there are too distant between the two conditions.

The table below introduces more accurate results achieved on the *lectmtg* task.

Table 3. Detailed results of the *lectmtg* task.

Condition	Full Miss	Miss Begin	Miss End	Miss In	False Alarm	Error Rate
adm	3,65%	3,06%	4,25%	12,20%	4,66%	27,81%
	(13,12%)	(10,99%)	(15,28%)	(43,87%)	(16,75%)	(100,00%)
mdm	5,83%	3,52%	5,00%	11,59%	6,65%	32,59%
	(17,89%)	(10,81%)	(15,34%)	(35,56%)	(20,40%)	(100,00%)
msla	4,52%	3,65%	4,69%	12,51%	5,24%	30,61%
	(14,75%)	(11,92%)	(15,33%)	(40,88%)	(17,12%)	(100,00%)
sdm	3,73%	4,16%	6,11%	13,93%	5,94%	33,87%
	(11,00%)	(12,29%)	(18,05%)	(41,13%)	(17,53%)	(100,00%)

The results of our SAD system over the *lectmtg* task are better than on the *confmtg*. In absolute, the overall error rate is 10% lower (~30% overall error rate). We can also remark that the percentages for all conditions are similar: in average 4.5% of *full miss*, 3.6% of *miss begin*, 5% of *miss end*, 12.5% of *miss in* and 5.6% of *false alarm*[3]. We can quickly see that the *miss in* errors represents a huge amount of the error rate (40%). *False alarm* and *miss end* follow with almost 20% of the errors. We can analyze these results saying that the data of the *lectmtg* evaluation are closer to the capabilities of our system than *confmtg*. We still see a huge amount of boundaries problems, lost intra speech segments remain our major problem. Finally, the good outcome is that our system can detect ~95% of the speech turns even if the boundaries are not precisely located.

At end, we can draw some conclusions on this evaluation. Our SAD system is not perfect if we look only at the final percentage. Most of the time, problems are boundaries (*miss begin*, *miss in*, *miss end*) and a non negligible part is *false alarm*. After looking at some labels and reference files, we can say that these problems seem to be less present at the end of the seminars. The adaptation process seems to refine the thresholds but with a too long latency. As our system is designed to run permanently, it is usually not a problem. During evaluation, the SAD system starts from scratch for each file. As we already said, we need transitions in order to compute new values. If we do not have enough transitions, it is obvious that we will not have a suitable adaptation process. For the next evaluation, we should consider to use our system in real condition but for this experiment, we decided that grouping seminar by collecting site, thus by similar recording equipments, settings and conditions, can be considered as cheating. We have chosen not to do so.

[3] As a comparison point, the false alarm rate for a system always answering *speech* is 25.44% on this task.

For us, the major result of our system for the RT06s evaluation is the low percentage of *full miss* speech turns (13% of the *mdm* and 1.6% for the *sdm* condition of the *confmtg* task and <5% in average for the *lectmtg* task). This score validates the usability of our SAD system in the context we are using it: detecting speech turns and people interactions for context modeling.

4 Conclusion and future work

Our speech detection system exposes performances that make it suitable for our projects and goals such as CHIL [11] or [12]. Actually, we do not want to use it for automatic speech recognition or diarization but for interaction and context modeling. Thus, we do not need precise speech boundaries but only to be sure to detect interaction between people so at least a part of each speech turn. If we look to the previous section, we can see that this goal is fulfilled: the *full miss* score is low. We think that this result is good for a generic system not trained on twin data of the evaluation that can run in different working environment (smart office, conference room, amphitheater, etc.) without any preliminary training/adaptation. Concerning our speed constraint, we saw that our system is successful because we can process many inputs in parallel. Following proposed improvements will be integrated carefully in order to preserve this capability.

For future work, experiments described in this paper have shown that some improvements of our system still need to be carried out. First of all, the online adaptation of the energy thresholds has to be refined. It could improve the performance of the system but decrease the system accuracy on adaptation failure. Next, we also envision extending the training of our neural network. We still want to keep an *a priori* training for this neural network while including different kind of speech recorded in different environment and different conditions (lapel and distant microphone). We think that doing this will provide us with a more generic speech detection system. We are currently preparing the corpus required by such extended learning.

We also want to improve our fusion scheme. We can substitute our rule based approach by a Bayesian one. The current expert-defined rules are rigid and could be advantageously replaced by a Bayesian fusion process. Moreover, if we plan to add other speech activity detection sub-systems, it will be difficult to rebuild a new set of rules. Doing this Bayesian training will also require having a learning corpus.

The last source of improvement would be to take advantage of all the available data in our perceptive environment. Our system could use visual information (facial features, visual clues of sound events, etc.) to improve the performance of speech detection.

5 References

1. D. Macho, J. Padrell, A. Abad, C. Nadeu, J. Hernando, J. McDonough, M. Wolfel, U. Klee, M. Omologo, A. Brutti, P. Svaizer, G. Potamianos, S.M. Chu. Automatic Speech Activity Detection, Source Localization, and Speech Recognition on the Chil Seminar Corpus. In IEEE International Conference on Multimedia & Expo, January 2005.
2. J. Ramirez, J. Segura, C. Benitez, A. de la Torre, A. Rubio. Efficient voice activity detection algorithms using long-term speech information. Eurospeech'97, pages, 1997.
3. A. Martin, D. Charlet, L. Mauuary. Robust Speech/Non-Speech Detection Using LDA Applied to MFCC. In Proc. ICASSP, vol. 1, 237-240, Salt Lake City, May 2001.
4. M. Frigo, and S.G. Johnson, The Design and Implementation of FFTW3, special issue on "Program Generation, Optimization, and Platform Adaptation", volume 95, pages 216-231, 2005.
5. J. Taboada, S. Feijoo, R. Balsa, C. Hernandez, Explicit estimation of speech boundaries, IEEE Proc. Sci. Meas. Technol., vol. 141, pp. 153-159, 1994
6. L. Rabiner, B.H. Juang, Fundamentals of Speech Recognition, Prentice Hall PTR, ISBN 0-130-15157-2, 1993.
7. L. Lamel, J.L. Gauvain, M. Eskenazi, BREF, a large vocabulary spoken corpus for French. In Proc Eurospeech'91, Genova (Italia), 1991.
8. D. Vaufreydaz, Modélisation statistique du langage à partir d'Internet pour la reconnaissance automatique de la parole continue, Ph.D. in Computer Science at Joseph Fourier University, Grenoble (France), 226 pages, January 2002
9. Spring 2006 (RT-06S) Rich Transcription Meeting Recognition Evaluation Plan, http://www.nist.gov/speech/tests/rt/rt2006/spring/docs/rt06s-meeting-eval-plan-V2.pdf.
10. F. Metze, J. Mc Donough, H. Soltau, A. Waibel, A. Lavie, S. Burger, C. Langley, L. Levin, T. Schultz, F. Pianesi, R. Cattoni, G. Lazzari, N. Mana, E. Pianta, L. Besacier, H. Blanchon, D. Vaufreydaz, L. Taddei, The Nespole! Speech-to-Speech Translation System, Human Language Technologies 2002, San Diego - California (USA), 6 pages, mars 2002.
11. F. Metze, P. Gieselmann, H. Holzapfel, T. Kluge, I. Rogina, A. Waibel, M. Wolfel J. Crowley, P. Reignier, D. Vaufreydaz F. Bérard, B. Cohen, J. Coutaz, S. Rouillard V. Arranz, M. Bertran,, H. Rodriguez, The "FAME" Interactive Space, 2nd Joint Workshop on Multimodal Interaction and Related Machine Learning Algorithms, Edinburgh - UK, 4 pages, February 2005.
12. O. Brdiczka, J. Maisonnasse, P. Reignier, Automatic Detection of Interaction Groups. In Proc. Int'l Conf. Multimodal Interfaces, October 2005.