FP7-ICT-2013-C FET-Future Emerging Technologies-618067

SkAT-VG: Sketching Audio Technologies using Vocalizations and Gestures

D5.5.1
Blind classifiers of imitations

First Author	Geoffroy Peeters
Responsible Partner	IRCAM
Status-Version:	Draft-0.5
Date:	December 1, 2015
EC Distribution:	Consortium
Project Number:	618067
Project Title:	Sketching Audio Technologies using Vocalizations and Gestures

Title of Deliverable:	Blind classifiers of imitations
Date of delivery to the EC:	30/11/2015

Workpackage responsible for the Deliverable	WP5
Editor(s):	Geoffroy Peeters
Contributor(s):	Geoffroy Peeters, Jules Françoise, Frederic Bevilacqua, Anders Friberg, Enrico Marchetto, Gabriel Meseguer Brocal
Reviewer(s):	Davide A. Mauro
Approved by:	All Partners

Abstract	The deliverable D5.5.1 presents the results of the tasks 5.1 and 5.2 of the WP5 of the SKAT-VG Project. We present the current results related to the automatic recognition of the sound categories defined in WP4 using the vocal and gesture signals of the imitations. We also present the current results of the automatic recognition of the vocal primitives defined in WP3 using the vocal signal of the imitations. The latter will be used in the next period for the construction of the informed classifier D5.5.2.
Keyword List:	

Disclaimer:

The SkAT-VG Consortium consists of the following entities:

#	Participant Name	Short-Name	Role	Country
1	Università Iuav di Venezia	IUAV	Co-ordinator	Italy
2	Institut de Recherche et de Coordination Acoustique/Musique	IRCAM	Contractor	France
3	Kungliga Tekniska Högskolan	KTH	Contractor	Sweden
4	Genesis SA	GENESIS	Contractor	France

Document Revision History Deliverable D5.5.1

Version	Date	Description	Author
v0.1	16/07/2015	Import from .tex template	D. A. Mauro
v0.2	10/09/2015	First version with contributions from all partners	G. Peeters, F. Bevilacqua, J. Françoise, A. Friberg, E. Marchetto
v0.3	14/10/2015	Draft version completed	G. Peeters, F. Bevilacqua, J. Françoise, A. Friberg, E. Marchetto
v0.4	15/10/2015	Minor template fixes and proofreading	DAM
v0.5	16/11/2015	Transition to final version	G. Peeters, F. Bevilacqua, J. Françoise, A. Friberg, E. Marchetto

Table of Contents

Index of Figures

List of Acronyms and Abbreviations

DoW Description of Work

EC European Commission

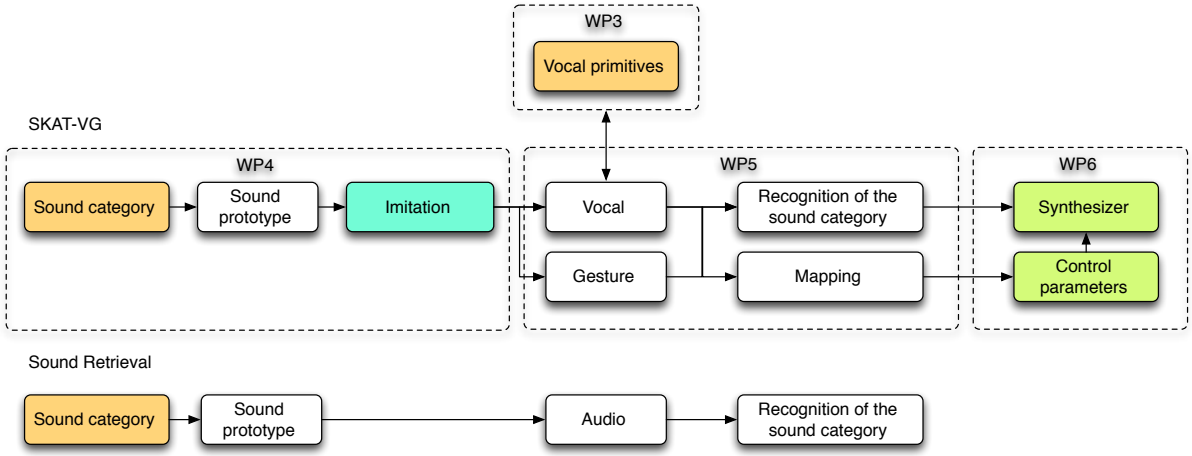
PM Person Months

WP Work Package

1 Executive summary

The scientific objective of WP5 is to investigate a class of signals that has been rarely studied: the non-linguistic vocalizations and gestures. WP5 develops new signal processing tools that are tailored to the specificities of these signals.

The application objective of WP5 is to recognize sound categories from the observation of the vocal/gesture signals of people imitating sound categories. It is farther reaching than simply recognizing them directly from the observation of their audio signal (as in a usual sound retrieval task).



The methodology used in order to achieve these objectives is to perform a deep analysis of the data-set provided by WP4. This data-set provides a large set of examples of how people imitate sound categories by voice and gesture. From this, we study the most appropriate representation of the non-linguistic vocalizations and gestures used for imitations. This representation can rely on the audio signal properties, on the underlying vocal mechanism (the vocal primitives proposed in WP3) or on the proposed gesture primitives. We then study the best possible model to represent the relationship between the vocal and gesture characteristics and the sound categories.

Parallel works in WP6 have allowed associating a synthesis (a set of synthesis) methods (and their associated control parameters) to each of the sound categories. The two applications of WP5 within WP6 are therefore

Recognition paradigm : to allow for the automatic selection of this synthesizer by voice and gesture (to know which are the most appropriate synthesizers and parameters to reproduce the sound category being imitated) and

Control paradigm : to allow controlling this synthesizer by voice and gesture (to allow a high-level control of those synthesizers through mapping of vocal and gesture).

During this first part of the project, WP5 concentrated on the recognition paradigm. This has been chosen in order not delay the development of the sound design tool. These results are presented in the current deliverable.

Now that a solid basis has been established both concerning the vocal/gesture signal representations and the usability of the recognition system, WP5 will next focus on the second goal: the control paradigm.

1.1 Main achievements

WP5 has started in December 2014. The first version of WP4 vocal and gesture imitation data-set was received in January 2015, and the final version in April 2015. The current status of WP5 is the following:

- The blind classifier using only the audio signal has been developed for the three sound families Abstract, Interaction and Machine.
- An efficient gesture representation based on Continuous Wavelet Analysis has been proposed and applied to the recognition of Abstract sounds and gesture primitives. This representation will be used for controlling the parameters of the synthesis.
- The automatic transcription system into vocal primitives has been developed and will allow the development of the task 5.3 "Informed Classifier".

1.2 Content of this deliverable

The deliverable is organized in three sections corresponding to the three main contributions performed so far by the partners in WP5:

Part 2 : IRCAM Sound-Analysis-Synthesis Team (SAS) has studied the automatic recognition of the Abstract, Interaction and Machine sound categories (defined in WP4) from the vocal signal of the imitations.

Part 3 : IRCAM Sound Music Movement Interaction Team (ISMM) has studied the representation into primitives of gestures and the automatic recognition of Abstract sound categories (defined in WP4) from the gesture signal of the imitations.

Part 4 : KTH has studied the automatic recognition of vocal primitives (defined in WP3) from the vocal signal of the imitations.

1.2.1 Automatic recognition of the sound categories from the vocal signal of the imitations

IRCAM SAS team has studied three different approaches to recognize the *sound categories* (defined in WP4) from the *vocal imitations*.

Description of the general approach:

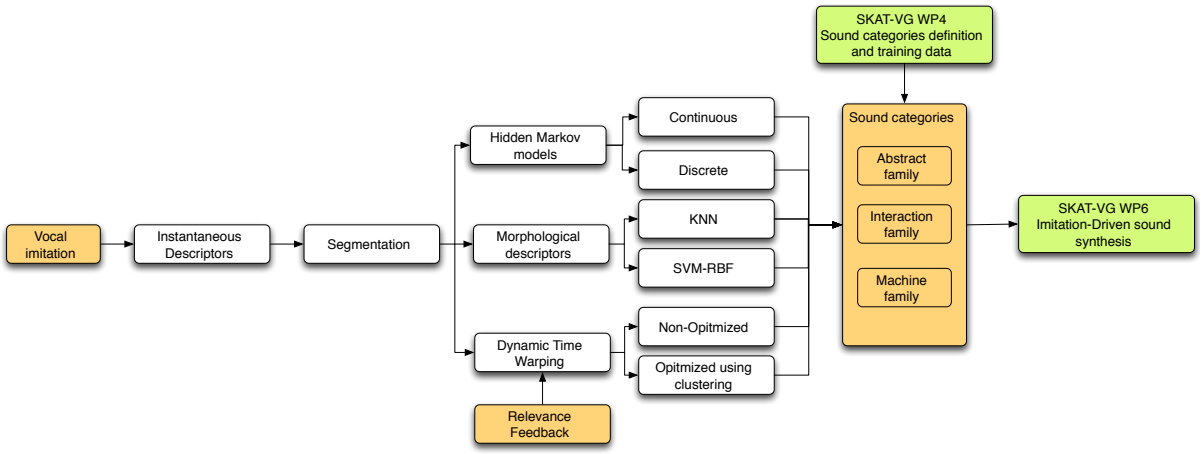
1. After discussions with the use case developers (WP6) it has been decided that the three families will be considered independently, as three possible point-of-views for the description of a vocal imitation. A given sound can therefore be described as a "Vehicles exterior" considering the Machine point-of-view and as "Up" considering the Abstract point-of-view. Therefore, three independent classifiers (into 6, 10 and 10 categories respectively) have been developed.
2. Also given the small size of the training/test data (down to 100 examples per class), it was also decided to remain in a low-dimensional feature space.

Approach 1 Given that the categories within the Abstract family mainly refer to the morphology of the temporal evolution of the audio content, we first tested two types of *hidden Markov model*: continuous (using directly the audio features as observation) and discrete (performing a quantization of the features into symbols).

Approach 2 Given the low recognition results obtained with the HMM approach, it was then decided to develop specific audio features to highlight the content of WP4 data-set. Those features, denoted by "*Morphological features*" aimed at representing the various specific aspects of vocal imitations (such as the use of voiced or unvoiced sound, resonance, the use of repetition). Two different set of morphological features have been developed: one dedicated to the Abstract family (AbsMorpho) and one generic applicable to the three families (GenMorpho). The GenMorpho audio features have been used in WP4 to perform a deep analysis of the content of the imitation data-set and to study the various strategies used by subjects to imitate sound. When used as input to a classifier (KNN or SVM) the GenMorpho audio features allow to correctly recognize the categories inside a given family with recall: 84.3% (Abstract), 71.6% (Interaction) and 69.0% (Machine).

Approach 3 We finally tested the use of direct time series comparison (using *Dynamic Time Warping*) to perform classification (using K-NN). The advantage of this approach is that it does not necessitate the development of dedicated audio features. Also it allows the inclusion of the relevance feedback mechanism that we studied in this report. The drawback of this approach is its computational cost. For this reason we studied an optimization technique relying on a pre-clustering of the time series. The aim was also to highlight the various strategies used by subjects. Unfortunately, this clustering was not successful.

This DTW approach (without optimization) leads to the best results (recall): 91.66% (Abstract), 71.55% (Interaction), 71.83% (Machine). These results should be compared to the ones of a random classifier for the equivalent number of categories: 16.66% (Abstract), 10% (Interaction) and 10% (Machine). We also remark that some categories are weakly recognized (such as "Filling" in the Interaction family or "Fridge", "PrinterFax", "Windshield" or "Vehicleint" in the Machine family) and should be filtered out when used in the sound design tool.



IRCAM SAS: automatic recognition of the sound categories from the vocal signal of the imitations using three approaches: hidden Markov models (continuous or discrete), morphological descriptors (using KNN and SVM RBF) and Dynamic Time Warping (without or with optimization by sequence clustering).

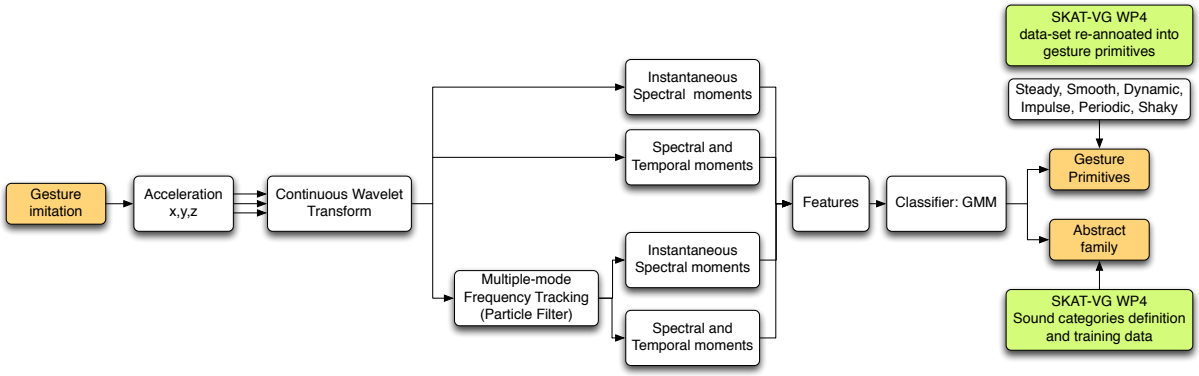
1.2.2 Automatic recognition of the sound categories from the gesture signal of the imitations

IRCAM ISMM team has studied the *gestures* used to imitate the different sound categories. By analyzing the gestures in the data set of vocal and gestural imitations (WP4), we proposed a set of gesture primitives representing the frequency modes of prototypical movements: steady, smooth, dynamic, impulse, periodic, shaky. These primitives have been manually labeled and found to consistent between two annotators in the Abstract sound family.

Considering the quantitative gesture analysis, we proposed to use a Continuous Wavelet Transform (CWT) of the acceleration parameters in the x, y and z axes. We found that the CWT gives in our case a better compromise time-frequency resolution compared to the Windowed Fourier Transform. Off-line and On-line (causal) versions of the CWT analysis have been implemented.

The analysis of the scalogram allows for extracting different gesture descriptors. A first set of instantaneous gesture features is derived from the spectral (scale) moments of the CWT. A second one (non-instantaneous) is derived from both temporal and spectral moments of the scalogram. A more sophisticated set of features is also derived by tracking the superposition of different ridges in the scalogram. The tracking is performed using Particle filter.

The system allows to automatically recognize the gesture primitives with a 66% accuracy and recognize the categories of the Abstract family (defined in WP4) with a 50% mean-Recall. While this is clearly too low to be directly applied in applications, this recognition rate is much higher than expected. This confirms, as hypothesized, that the Wavelet based descriptors allows for capturing some frequency modes that are consistent among several participants. After some refinement, we thus expect that our general approach will be suitable to be applied in the SkAT-VG applications.

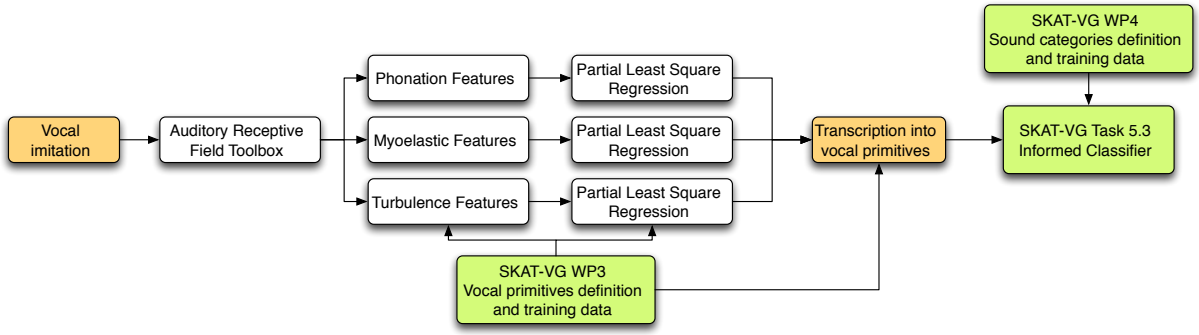


IRCAM ISMM: automatic recognition of gesture primitives and Abstract sound categories from the gesture signal of the imitations.

1.2.3 Automatic recognition of the vocal primitives from the vocal signal of the imitations

KTH has studied the automatic transcription of an vocal imitation into *vocal primitives*. Among the vocal primitives studied in WP3, the following three were chosen: phonation, myoelastic and turbulence. Each of them is predicted using an independent binary classifier: phonation/no-phonation, slow-myoelastic/no-myoelastic, turbulence/no-turbulence. Auditory Receptive Fields (ARF) are used to represent the audio signal. It should be noted that this research is the first one to apply the ARF to a practical problem. From the ARF, three set of features have been developed to represent the three problems. These features are then used as input of a Partial Least Square Regression classifier. Using a leave-one-subject-out evaluation (with 4 subjects), the system obtains the following accuracies: 92.0% (phonation), 84.6% (myoelastic) and 77.7% (turbulence).

This automatic transcription of a vocal imitation into vocal primitives will be used as input for the development of the next task 5.3 "Informed Classifier".



KTH: automatic recognition of the vocal primitives from the audio/vocal signal of the imitations.

1.2.4 Description of WP4 data-set of vocal and gesture imitations

In WP4, a total of 26 sound categories have been defined organized in three families: Abstract (6 categories), Interaction (10 categories) and Machine (10 categories). Each category c is

represented by two reference sounds r. Each reference sound has been imitated by 50 subjects s, and each subject could use several trials t. The subjects did the imitations in two distinct setups: using only the voice (Voice Only, VO) and then using both voice and gesture at the same time (Voice plus Gesture, VG).

Scientific questions raised by the data-set

1. *Do subjects imitate the sound categories c or the specific reference sound r chosen as an exemplar?*

   The answer to this question should allow us to decide whether we need to develop a system to recognize the categories c or directly the reference sounds r. The answer to this question comes from the analysis of the confusion matrices of the systems trained to recognize reference sounds r. From these matrices (see Figures 8, 9, 10), we can see that: (1) There are reference sounds from the same category (such as "stable1" and "stable 2") that are so specific that the strategy used by subjects to imitate them is very different; (2) There are reference sounds (such as "up1" and "up2") that are not specific such that the strategy used by subjects to imitate them is close.
   The grouping of reference sounds r into categories c should therefore be done on a case by case in the final recognition model.
   Of course, one could train a system to always recognize the reference sounds r and then perform a hierarchical grouping to reach the category level c; however training directly on the reference sounds r leads to a decrease of the number of training example (down to 50).

2. *Do all subjects s use the same strategy to imitate a category c? Is the difference between the strategies used by different subjects s to imitate the same category c larger than the one used by a single subject s across categories c?*

   To answer this question, we performed the analysis of the whole data set using the GenMorpho audio features presented in part 2. The results, published in the deliverable D4.4.1 (part 4.1.3) indicate that whereas there exists some marginal clusters of subjects in terms of strategy, there is no clear clustering in terms of strategy used by subjects.

3. *Do subjects s use the same strategy over trials t?*

   In the current draft version of this deliverable we do not have yet quantitative results to answer this question In some of our classification tests, we used only the last trial t of each subject considering that this should represent the final choice of the subject. However doing this reduces the number of training/testing data per class to 100. In some other experiments, we considered the whole set of trials t. In the table 11, 12 and 13, we compare the results obtained using only the last trial and using all trials. While for Abstract and Interaction categories, using only the last trial provides a higher recognition rate, this is not the case for the Machine categories.

   Related to this point, it is important to mention that in all our experiments we applied a subject-independent evaluation, i.e. a subject s is never part simultaneously of the

training and testing data.

1.3 Future works

The future works concern the following points

- to define a general methodology to define vocal and gesture primitives
- evaluate multi-modal (audio and gesture) sound categories recognition
- development of the informed classifier based on vocal primitives recognition
- integration of a relevance feedback mechanism
- integration of blind classifiers in the sound design tool (WP6)
- development of mapping strategies between vocal/gesture to synthesis parameters

2 Audio analysis and recognition

2.1 Overview

There is a very broad range of approaches that can be applied to do blind classification on vocal signals. In the following sections, we present the path that we have followed and the main results which have been obtained so far.

The audio content of the SkAT-VG dataset is completely made of vocalizations. Our first step has thus been inspired by a common approach used in automatic speech recognition. We can not rely on any prior assumption on the language structure, hence our methodology has been rather simple: low level audio features are modeled by Hidden Markov Models (HMMs, sec. 2.5). Two specific sets of audio descriptors have been developed. A first one extracts the local information of the audio signal, and is used with continuous HMMs. Then, we introduce a set of descriptors which extracts long-time characteristics from the signal, and model it using discrete HMMs.

After having obtained some results with this methodology, we moved to a novel approach based on Morphological descriptors (sec. 2.6). The characteristics of the imitations, especially the almost absent linguistic content, prompted for an innovative way of describing the vocal signal, geared toward the inner acoustical cues of the recordings. Our development started from the Abstract family, in which each category is defined on some expected signal content. The obtained results were good, thus we applied the same methodology to develop a more general set of morphological descriptors. The latter has been successfully applied to the three families, obtaining good results with minor per-family tunings.

Our research then focused on a set of descriptors which are not geared toward imitation recognition only. The motivation of our effort has been the development of a methodology with a more general scope, which can be of scientific interest. The two previous approaches share a common step: the low level signal descriptors are resumed in a compact form. By contrast, in sec. 2.7 we propose the use of the original descriptor time series, compared each other by means of a suitable distance. This approach leads to results which are comparable to the ones obtained by morphological descriptors. Moreover, the direct use of time series will enable a straightforward implementation of relevance feedback, as introduced in sec. 2.8.

2.2 Dataset preprocessing

In this section we deal with the audio recordings of the SkAT-VG project dataset, which is actually multimodal (see D4.4.1). Unfortunately, we have verified that the available sounds are not well suited for direct use in classification tasks, because recording artifacts are often present.

As part of Task 4.2 of WP4 (sec. 3.2 of D4.4.1) a manual annotation of the dataset has been carried out, by listening to all the sounds. Those which have been clearly identified as non pertinent to the requested imitation have been deleted. We point out that, to keep all the dataset variability, no *strange* imitation have been deleted, but only recordings with technical problems. For each kept file an annotation has been provided, using the XML format; these annotations clearly mark the begin and end times of the imitation stored in the corresponding audio file.

The files provided by WP4 are thus all annotated, but still not ready for automatic classification. We have found that in many cases the audio/video time references were present, resulting in a strongly audible *click* on the audio files. An automatic de-click procedure has thus been applied to the dataset: the click noise position in time (if present) has been precisely found by autocorrelation with original click sample, then to remove it a sample-by-sample subtraction has been applied.

The last step before actual development of recognition has been the extraction of metadata. The dataset has been collected using a number of experimental conditions, thus resulting in a lot of information being related to each sound file. All the data is available in the original files names, thus we have applied an automatic parser to decode the metadata from each file name and organize it in a table format, along with annotation information. Some software routines then enable an easy subset selection, based on given criteria.

2.3 Low-level features

Once the dataset has been prepared, the audio signals are ready to be processed in order to extract a set of low-level audio features. These will then be used in all the following steps.

From each audio file in the dataset, 7 instantaneous audio features $d_i(k)$, $i \in [1, ..., 7]$ are extracted, where k denotes the time frame number. In order to smooth the variation of $d_i(k)$ over time, a low-pass filter is applied (zero-phase filter). The first 5 features are standard: loudness, spectral centroid, spectral spread, spectral rolloff and pitch. They are computed using well-known techniques [Pee04]. The pitch is computed using the Swipep algorithm [CH08]; we used this since it has state-of-the-art performances and is readily available on-line.

We also propose two new audio features: "LPC-min" and "Spectral-peak-min". The novel features are defined as follows:

LPC-min A one-pole pre-emphasis filter is applied. Low-order LPC is used to estimate the position of the single most prominent formant. The objective is not a full-fledged formants tracking, but a robust analysis of the energy location among frequencies, complementary to spectral centroid. The prediction coefficients are converted into formant frequencies F_ρ, where ρ is the formant index [McC74, MG76]. Only the frequencies $F_\rho > 20\mathrm{Hz}$ are kept. The LPC-min value is measured in Hz, and is defined as the minimum F_ρ.

Spectral-peak-min From the energy spectrum (computed as the square DFT) we select the 5 most important frequency bins. The Spectral-peak-min is defined as the lowest frequency among these 5 frequencies; it is thus measured in Hz. The described algorithm is naive and is not immune from discontinuities, even very frequent, in the output frequency cue. However many empirical tests showed that, taking only the topmost 5 points in a 1024 set and using a window with a rather large main lobe, the routine gives an almost continuous cue. The obtained envelope follows the frequency in the signal associated with highest energy, and is conceived to cope with signals which can be harmonic or not. In harmonic sounds the resulting envelope approximates the pitch, while with inharmonic input the envelope works like the spectral centroid.

It should be pointed out that these features could have overlapping meanings, and are used together to reinforce the information. Spectral centroid and LPC-min could be similar on noisy sounds, but when a strong formant is present the centroid may loose meaning compared

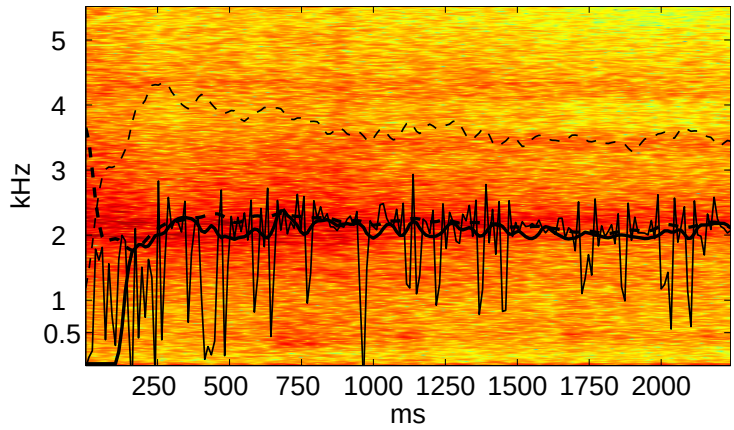


Figure 1: Comparison between LPC-min (dashed bold line), Spectral-peak-min (bold line), spectral centroid (dashed line) and pitch by Swipep (thin line). This vocal imitation is noisy with a stable formant around 2kHz. Swipep does not detect any pitch, giving unreliable information, and the spectral centroid is moved toward higher frequencies. Both LPC-min and Spectral-peak-min are instead detecting and following the formant.

to LPC-min (Fig. 1). Spectral centroid and Spectral-peak-min could also be similar on noisy signal, but when a strong partial exists at the pitch the Spectral-peak-min is better at measuring it (Fig. 2).

2.4 Segmentation

During the extraction of the low level features, the signals are analysed for silent/non-silent regions; we denote non-silent ones as Active Regions. For each audio file, along with the features time series, the analysis provides a set $A = \{[b_r, f_r] : r \in [1, ..., N']\}$ which holds the begin(s) and the end(s) times of the Active Regions. This automatic segmentation phase should not be confused with the manual annotation of the dataset (sec. 2.2): in this case, many different Active Regions can be present in the same recording.

2.5 Recognition using hidden Markov models

Once the low level features described in sec. 2.3 have been extracted from all the recordings, it is necessary to resume their main characteristics in order to exploit them by machine learning tools. In this section we present a first approach to do this: the features time series are modified such to enhance their informative content. The time evolution of the signals is then represented by suitable tools, such as Hidden Markov Models (HMMs).

We have defined two sets of descriptors: the Local Trend, to be used with continuous HMMs, and the Global Trend, modeled by discrete HMMs.

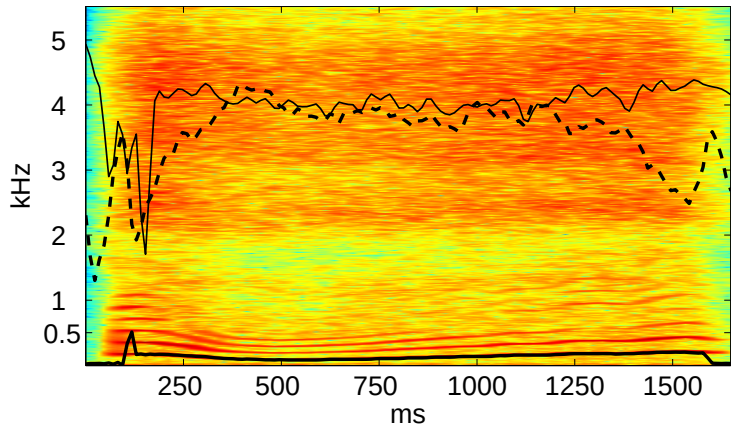


Figure 2: Comparison between LPC-min (bold line), spectral centroid (dashed bold line) and Spectral-peak-min (thin line); pitch is not reported because perfectly matches LPC-min. This vocal imitation is harmonic but presents also noise in higher frequencies. LPC-min is clearly detecting the pitch; Spectral-peak-min is following the energy of noise, with better accuracy than centroid.

2.5.1 Continuous HMM

The six categories of the Abstract family relate to evolution of values over time, hence we compute the derivative of each low level feature $d_i(k)$ (except Loudness). The derivative $d'_i(k)$ is found by linear regression on the local values (5 points on the left and 5 points on the right of k). We then normalize $d'_i(k)$ range to $[-1, 1]$ using arctangent mapping: $d''_i(k) = 2/\pi \arctan(d'_i(k))$. The computation of Local Trend descriptors is thus complete; an example of these is shown in Fig. 3.

As anticipated before, explicit time modeling has been achieved using HMMs. For each of the six categories c of the Abstract family, a continuous HMM $\mathcal{M}_c$ is defined. Each $\mathcal{M}_c$ represents the transition between a set of $S = 4$ states. The emission probability (probability of emitting $d''_i(k)$ given state s) is modeled as a mixture of $M = 8$ Gaussians and diagonal covariance matrix. The HMMs are created in a supervised way. To understand this, let's consider the case of the `up/down` category. This one can be represented as the transition from a state "silent" to state "up" to state "down" and back to state "silence". In the same spirit the `up` category can be represented as the succession of states "silence", "up", "silence". We therefore define 4 states: "silence", "up", "down", "stable". The training of the six HMMs is done in two stages:

- We first train the observation probabilities. This is done independently of $\mathcal{M}_c$. Indeed, given that a state (such as "up" in the above example) can be shared by different $\mathcal{M}_c$, we train the emission probabilities using descriptors from the `up`, `down` and `stable` categories, plus an added silent class.
- The transition probabilities are trained for each category.

Considering that the number of self-transition $s(t+1) = s(t)$ is much larger than the non-self ones $s(t+1) \neq s(t)$, we found the training of the last difficult. In order to circumvent

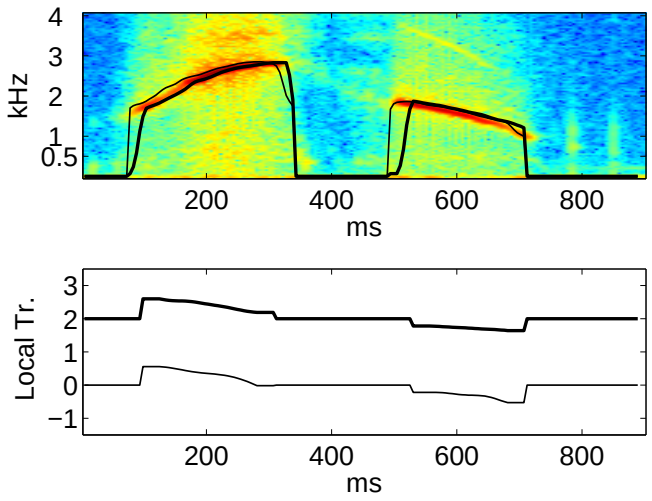


Figure 3: Example of descriptors on `up/down` imitation. Topmost panel is the signal spectrogram with spectral centroid (thin line) and Spectral-peak-min (bold line). The bottom panel shows the related descriptors (scale is shifted for Spectral-peak-min for clarity).

this, we decided to decimate over time the descriptors time series by a factor of 3. This allowed to increase the performances. Also, rather counterintuitively, better results were obtained by forcing the HMM training to *not* update the emission probabilities mixtures (thus only updating the transition matrix).

2.5.2 Discrete HMM

The same audio features of sec. 2.3 are used as underlying time series for another set of descriptors, used with discrete HMMs: spectral centroid, spectral spread, spectral rolloff, pitch, LPC-min and Spectral-peak-min (not Loudness). Instead of using them directly as input to an HMM, we convert them into symbols.

For each descriptor $d_i(k)$ and each Active Region r we apply the following procedure:

- We compute the linear regression over the region r. We denote by α_i^r its angular coefficient and by ϵ_i^r its prediction error.
- If ϵ_i^r is larger than a specific threshold K_1[1], r is split into two regions at the position of the maximum value of $d_i(k)$ and a two-piece minimum least-square linear regression is computed. We denote by α_i^{r1} and α_i^{r2} the corresponding angular coefficients.
- The quantized time serie $e_i(k)$ corresponding to $d_i(k)$ is then built. It has the value $e_i(k) = 0$ for k corresponding to silent part, $e_i(k) = \alpha_i^r$ for k corresponding to region r (or α_i^{r1} for region $r1$ and α_i^{r2} for region $r2$).

[1] K_1 has been optimized by grid-search. We use a value of $3 \cdot 10^3$.

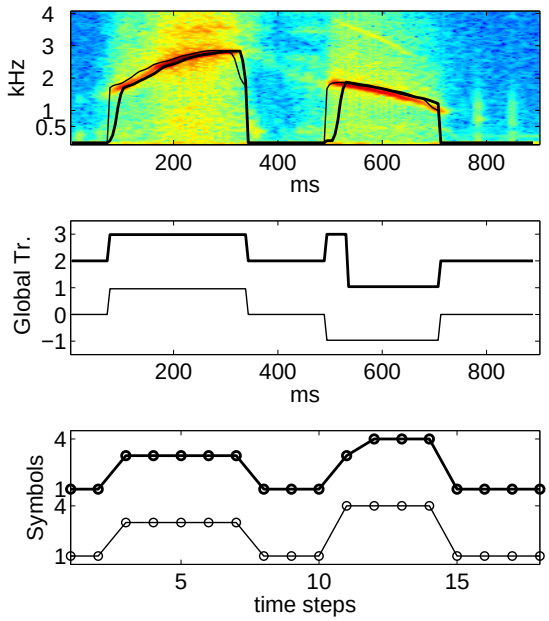


Figure 4: Example of descriptors for continuous HMMs on `up/down` imitation. Topmost panel is the signal spectrogram with spectral centroid (thin line) and Spectral-peak-min (bold line). Middle panel shows the two related descriptors (scale is shifted for Spectral-peak-min for clarity). In the bottom panel final results of the descriptors are shown: values are quantized into symbols and downsampled (once more Spectral-peak-min has been shifted).

The `up/down` category will be better described by the two-pieces regression than by a single regression; this will contrast with monotonic categories (such as `up` and `down`) thus enhancing the discrimination.

We then convert the values of $e_i(k)$ to symbols using the following rules:

$$e'_i(k) = \begin{cases} 1 & \text{if } |e_i(k)| \leq 10^{-7}; \\ 2 & \text{if } 10^{-7} < |e_i(k)| \leq 0.1; \\ 3 & \text{if } e_i(k) > 0.1; \\ 4 & \text{if } e_i(k) < -0.1. \end{cases} \quad (1)$$

The four symbols $\{1,2,3,4\}$ express the overall condition of the time serie, respectively: silence, stable (small angular coefficient), upward (large positive angular coefficient), downward (large negative angular coefficient).

As a final step, the descriptors series $e'_i(k)$ are decimated over time by taking one value every 10 frames. It should be noted that we choose to not apply any antialiasing process, since we found by experiment that Active Regions shorter than 10 samples are in large majority negligible. In Fig. 4 we illustrate the Global Trend descriptors.

Training Since the time series $e'_i(k)$ are symbols (unordered values) we model them using discrete HMMs. Each descriptor i is modeled by its own HMM $\mathcal{M}_{c,i}$.

For a given category c and a given descriptor i we denote by $E_{c,i}$ its emission matrix (with size $S \times O$, where S is the number of hidden states and O the number of symbols), by $T_{c,i}$

the transition matrix (with sizes $S \times S$) and by $S_{c,i}(k)$ the decoded states at frame k.

In order to train the emission matrix for category c and descriptor i, we concatenate all descriptors of the sounds belonging to c into a matrix D_c. Each row of D_c corresponds to a given descriptor i.

We define a function $\mathcal{F}$ that normalizes an input matrix such that its rows elements sum up to 1. The training algorithm is the following:

1. Set the initial value for the emission matrix to $E_{c,i} = \mathcal{F}(I + 0.05)$, where I is the diagonal identity matrix, with size $S \times S$. This almost associates each state to a single symbol, but does not exclude the possibility for each state to emit each possible symbol.

2. Exploiting the previous association, estimate $T_{c,i}$ by accumulating all the emissions transitions found in row i of D_c into the matrix T. Obtain the transition matrix for i as $T_{c,i} = \mathcal{F}(T)$.

3. Use $T_{c,i}$ and $E_{c,i}$ to estimate the hidden states $S_{c,i}(k)$ by Viterbi decoding.

4. Re-estimate $E_{c,i}$ by counting the number of times each state generates each emission, and again normalizing by $\mathcal{F}$.

In principle, the re-estimation procedure (points 2.-4.) can be iterated, but early experiments showed little performance improvements.

Classification In order to classify an unknown sound represented by its time series matrix D_*, we decode each row i of D_* using a specific category model $\mathcal{M}_{c,i}$ and the Viterbi decoding algorithm. Each model $\mathcal{M}_{c,i}$ provides a likelihood $l_{c,i}$. The final category label is found as $x = \text{argmax}_{c \in [1,6]} \sum_i l_{c,i}$.

2.5.3 Recognition results

Methods	Continuous HMM		Discrete HMM	
Measures	Rec.	Prec.	Rec.	Prec.
`up`	80.8	83.9	83.2	81.8
`down`	**88.5**	43.9	76.2	76.3
`up/down`	38.2	53.0	39.1	57.2
`impulse`	25.3	54.1	60.3	79.1
`repetition`	29.5	35.5	78.0	62.4
`stable`	72.9	99.2	**97.2**	70.0
Average	55.9	61.6	72.3	71.1
Avg. Accuracy	55.1		70.8	

Table 1: Recognition results of the **Abstract** family using HMM (averaged over 5 crossvalidation folds).

The descriptors introduced in sec. 2.5.1 and 2.5.2 have been applied to automatic recognition, and results are presented here. All the figures have been obtained using 5-folds crossvalidation, selecting train and test set in order to not have the same subject in both. The whole

set of Abstract family recordings has been used, disregarding the different sessions and trials (see D4.4.1). The results are reported in Tab. 1. Recall and precision values are given for each of the two methods and for each class. The mean recall and precision, and the overall accuracy, are given at the bottom of the table.

The Global Trend descriptors obtain an accuracy of 70.8%, a clear improvement if compared to the Local Trend accuracy of 55.1%.

It is interesting to look at the results class-by-class. `up` and `down` categories are well recognized by all methods. For `down` the best recall is given by Local Trend, but with poor precision. It has been verified that `up/down` recordings are often recognized as `down`, explaining the low precision. A possible motivation for the confusion arises by observing the spectrograms of the recordings: it has been found that many subjects prepare the downward slope in `down` by first producing a rising profile. Moreover, while `up/down` is recognized as `down`, the confusion with `up` is less frequent: in `up/down` the downward phase is usually stronger, thus justifying the observations.

Categories `impulse` and `repetition` are problematic for Local Trend: the transition matrices of the HMMs do not succeed to capture the temporal cues of the signals, and this has a bad influence on categories which are defined relying on that. By contrast, using the Global Trend the descriptors are quantized in four symbols, with the effect of keeping only the most relevant information; moreover the decimation enforces the transition matrices to describe the temporal cues of the signals. To confirm this, Global Trend outperforms Local Trend exactly by the improvement in the recognition of `impulse` and `repetition`.

2.6 Recognition using Morphological descriptors

The six categories of the Abstract family share a relevant property: the name of the category describes by itself the expected main characteristics of the imitation signals. This has brought our development toward the direct measure of these characteristics, without the need of an automatic learning tool to find them out. A first outcome of this effort is presented in sec. 2.6.1: a set of 8 scalar descriptors is defined. These have been developed for the Abstract family only, and are called **Abstract Morphological (AbsMorpho)**. This set of descriptors, along with the Local and Global Trend ones (introduced in sec. 2.5), have been the objects of a publication [MP]. The AbsMorpho descriptors obtain good classification results, but their use is limited to the Abstract family only.

We have then developed a second set of descriptors, which we call **General Morphological (GenMorpho)**, which can be applied to all the three families (sec. 2.6.2). This specific set of descriptors has been used, with some minor modifications, in Task 4.3 of WP4 to obtain a general overview of the dataset; the results of this analysis are reported in D4.4.1. The cited analysis has provided enough detail on the dataset contents, proving the effectiveness of the GenMorpho descriptors.

In sec. 2.6.3 are presented the automatic classification performances obtained with both types of Morphological descriptors. Unlikely the Local and Global Trend sets, the morphological descriptors embed directly the time evolution of the signal and there is no need of temporal modeling to do classification. Therefore, we use Support Vector Machines (SVM) as machine learning tool instead of HMMs. The GenMorpho set proves itself effective on the three families and is an important contribution of WP5.

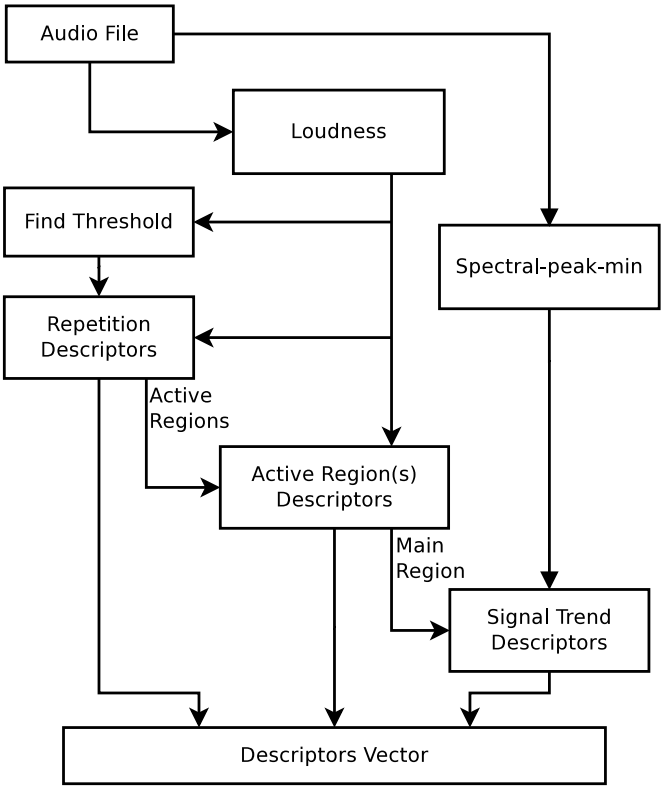


Figure 5: Computation of the AbsMorpho descriptors.

2.6.1 The AbsMorpho descriptors

We introduce here a first set of morphological descriptors tailored to the Abstract family only. These are crafted to compactly resume the structure of the signals present in the dataset. Each audio file is represented using a vector with 8 components:

- Ψ_1, Ψ_2 and Ψ_3 measure repetitions or patterns in the serie;
- Ψ_4, Ψ_5 and Ψ_6 describe the Active Region(s);
- Ψ_7 and Ψ_8 are related to the global signal trend.

Fig. 5 points out the main steps of the computation.

We recall here the low-level features introduced in sec. 2.3: as shown in Fig. 5, only Loudness and Spectral-peak-min are used. Moreover, the Active Regions A are available for each recording.

For two consecutive active regions of a signal, r and $r+1$, we define the duty-cycle of r as $u_r = (f_r - b_r)/(b_{(r+1)} - b_r)$ (see Fig. 6). In this, we assume that every active region is followed by a silent one. When adjacent regions are in the same state (silent or non-silent) we simply merge them, thus enforcing the active/non-active alternation.

• **Ψ_1, Ψ_2 and Ψ_3: measurement of repetitions or patterns in the serie.**

The first two pattern descriptors Ψ_1 and Ψ_2 are defined as the mean and the standard deviation of non-silent regions duty-cycles u_r, $r \in [1, N']$. `impulse` and `repetition` categories are expected to have small values of Ψ_1. In the opposite, the other categories (`stable`, `up`, `down`, `up/down`) will have a single long active region with a large value of Ψ_1. Ψ_2 measures the regularity of the repeated patterns. In the case of a single active region, $\Psi_2 = 0$.

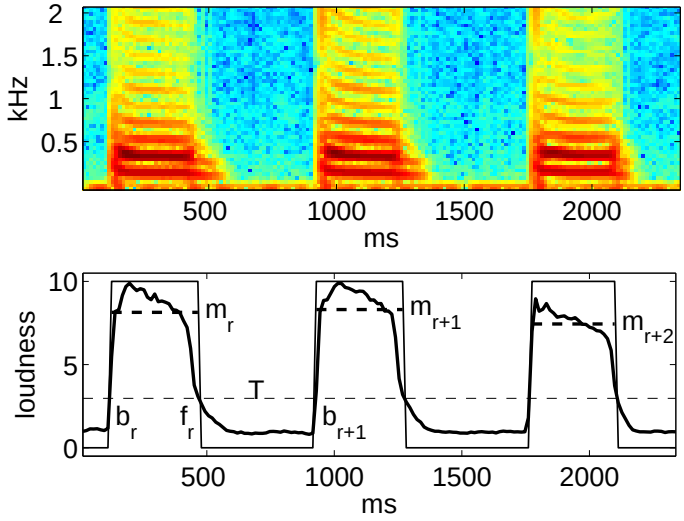


Figure 6: AbsMorpho descriptors: computation of Ψ_1, Ψ_2 and Ψ_3 on a `repetition` imitation. The loudness profile is showed (bold line), along with threshold T (dashed line) and active region detection (thin line); mean loudness value m_r (dashed bold line) is used to compute importance i_r.

We define the *Importance* i_r of an active region as the product between its length l_r and its mean loudness m_r (over the active region duration). Both l_r and m_r are normalized in the range $[0, 1]$ for each signal (the value of 1 is assigned to the longest and the loudest regions respectively).

The third descriptor Ψ_3 is the number of active regions which have *Importance* i_r above a threshold K_2. A value of $K_2 = 0.25$ is chosen, which corresponds to the product of half-range normalized values of l_r and m_r. Ψ_3 is computed as:

$$\Psi_3 = \frac{\arctan\left(\mathrm{card}\left(\{i_r \text{ such that } i_r > K_2\}\right) - 1\right)}{(\pi/2)} \tag{2}$$

The threshold on i_r allows the rejection of very short active regions, or with very low loudness level. For single-region signals Φ_3 is equal to 0, while for signals with three or more regions (typical of `repetition` category) Ψ_3 is above 0.7.

• **Ψ_4, Ψ_5 and Ψ_6 describe the Active Region(s).**

Descriptors Ψ_4 and Ψ_5 focus on the main region R only, that is the non-silent region with highest importance i_R.

The Ψ_4 descriptor is the duty cycle of the main region *defined on the whole signal*: $\Psi_4 = (f_R - b_R)/N$, where b_R and f_R are the start and the end of R and N is the total signal length. Ψ_4 is expected to discriminate `impulse` and `stable` categories.

The descriptor Ψ_5 is computed on the loudness time serie in the main region $d_L(k)$, $k \in [b_R, f_R]$, which has length l_R. The original serie and its half-length circularly-shifted copy are used:

$$\Psi_5 = \sum_{k=1}^{l_R} \left[d_L(k) - d_L\left((k + \frac{l_R}{2}) \bmod l_R \right) \right]^2 \quad (3)$$

Ψ_5 is thus the energy of the difference between $d_L(k)$ and its shifted copy. The descriptor is then normalized in the $[0, 1]$ range using $\arctan$, as for Ψ_3. Ψ_5 improves the discrimination between categories which have flat or non-flat evolution, such as `up/down` *vs* `stable`.

The descriptor Ψ_6 is defined as the sum of the active regions lengths l_r, minus a constant γ. Ψ_6 is then normalized by arctangent, such as in (2), and γ is optimized to have the "shift" of the arctangent function improving the discrimination between `impulse` and `stable` (or `repetition`) categories.

- **Ψ_7 and Ψ_8: description of the global signal trend.**

The Ψ_7 and Ψ_8 morphological descriptors have been developed to measure the slope of the signal. The aim is to evaluate the slope between the beginning and the middle, and between the middle and the end, of a given time serie.

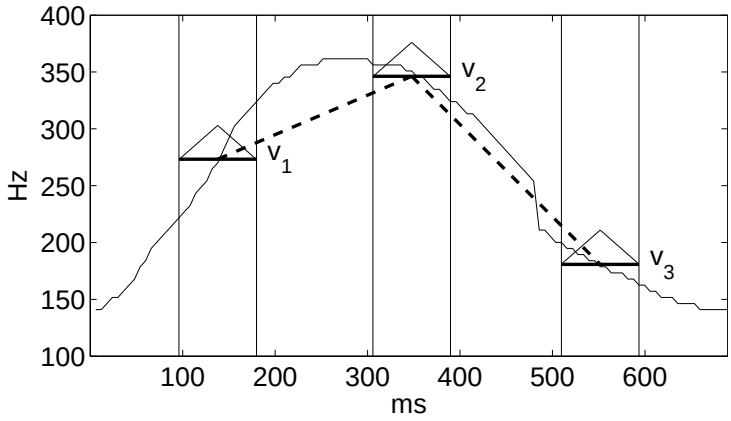


Figure 7: AbsMorpho descriptors: computation of Ψ_7 and Ψ_8 on an `up/down` imitation. Spectral-peak-min is showed (thin line), with 3 windows centered at $1/5$, $1/2$ and $4/5$ of its total length. Mean values v_j (bold line) are used to find Ψ_7 and Ψ_8, which are proportional to the slopes (dashed bold line).

Spectral-peak-min is taken as an underlying descriptor: only its values in the main region R are considered, deleting those below 40Hz as they are unreliable. To overcome boundary effects, Spectral-peak-min is observed within three windows of 11 samples taken at the beginning, the middle and the end of the region (see Fig. 7).

In each window, the Spectral-peak-min is weighted by a triangular window function and then the average is computed. This leads to three mean values: $V = [v_1, v_2, v_3]$. The trend

descriptors Ψ_7 and Ψ_8 are found as:

$$\Psi_7 = \frac{v_2 - v_1}{v_1} \qquad \Psi_8 = \frac{v_3 - v_2}{v_2} \tag{4}$$

and normalized using again the $\arctan$ function. Local windows are used in order to smooth the signal, possibly generated by noisy time series, and the triangular functions give more importance to the central part of the window. Ψ_7 and Ψ_8 measure the evolution in time of the signal: they discriminate between `up`, `down` and `up/down`.

2.6.2 The GenMorpho descriptors

The definition of morphological measures for the categories belonging to Machine and Interaction families is not straightforward. Differently from what happens with Abstract family, in this case we can not rely on expected category characteristics. The General Morphological (GenMorpho) descriptors, which we present in this section, have been conceived with this aim: to characterize the morphology of a recording from any of the three families.

To compute the GenMorpho descriptors, some low level audio features are extracted from each recording: loudness, total energy, noisiness, zero crossings, spectral centroid, spectral rolloff, pitch and pitch strength, Spectral-peak-min and LPC-min. Moreover, the Active Regions boundaries $A = \{[b_r, f_r] : r \in [1, ..., N']\}$ are found. This is similar to what already described in sec. 2.3: the references for the computation of these time series can be found there.

The GenMorpho descriptors consists of scalar values $\Phi_i, i \in [1, 13]$, computed as follows.

• **Φ_1, Φ_2 and Φ_3: description of the signal structure.**

The first descriptor Φ_1 is found as the number of Active Regions in the signal. Φ_2 and Φ_3 are defined, respectively, as the Absolute Duration and the Relative Duration descriptors. Φ_2 is the number of analysis frames in the signal which are marked as active (i.e. which are non-silent). Φ_3 is the ratio between the mean number of active frames of each Active Region and the total number l of frames (silent and non-silent) of the signal.

• **$\Phi_4 \dots \Phi_{10}$: description of average signal content.**

The descriptors $\Phi_i, i \in [4, 8]$ are defined as the median value of the following time series, respectively: noisiness, zero crossings, pitch strength, pitch and loudness. The descriptors are computed looking only at the frames belonging to Active Regions. Similarly to these, $\Phi_i, i \in [9, 10]$ are the standard deviation values of the pitch and spectral centroid time series, respectively. Once again, only active frames are used.

• **Φ_{11}, Φ_{12} and Φ_{13}: description of signal evolution.**

The following descriptors resume the temporal evolution of the signal. They are thus found by looking at the *slope* of some underlying time series, found in the following way. The whole time serie is considered, disregarding the Active Regions boundaries. Given an input signal, with duration l, two rectangular windows are applied on it: the first between $\frac{l}{10}$ and $\frac{l}{4}$, and

the second between $\frac{3l}{4}$ and $\frac{9l}{10}$. The mean values of the serie in the two windows are called v_1 and v_2, and the *slope* value is found as:

$$sl = \frac{v_2 - v_1}{l}. \tag{5}$$

This is equivalent to the angular coefficient of a linear interpolation along the whole signal length, and the two windows increase the noise robustness (we recall Ψ_7 and Ψ_8 from sec. 2.6). Φ_{11} is the Magnitude Slope, and is defined as the slope, computed by the above procedure, of the loudness time serie. Similarly, Φ_{12} is the Maximum Frequency Slope; it is found by computing the slopes of the pitch, spectral centroid, spectral rolloff, Spectral-peak-min and LPC-min series, and then taking the value which has maximum absolute value[2].

The last descriptor is complementary to Φ_{11}, as it measures how much the loudness time serie actually differs from the linear interpolation. The values v_1 and v_2, found while computing Φ_{11}, are used to approximate the loudness time serie $d_L(t)$ by the linear model $\hat{d}_L(t)$. The descriptor Φ_{13} is found as:

$$\Psi_{13} = \frac{\sqrt{\sum_{t=1}^{l} \left[d_L(t) - \hat{d}_L(t) \right]^2}}{l}. \tag{6}$$

2.6.3 Recognition performances

Recognition of the Abstract family using the AbsMorpho descriptors.

The values of the AbsMorpho descriptors, introduced in sec. 2.6.1, are normalized such that they lay in homogeneous ranges: because of this we can use for classification a k-Nearest Neighbor algorithm with Euclidean distance. It has been chosen to have $k = 5$, after having verified the effectiveness of this value by grid search.

All the figures have been obtained using 5-folds crossvalidation, selecting train and test set in order to not have the same subject in both. The whole set of Abstract family recordings has been used, disregarding the different sessions and trials. The results are reported in Tab. 2. Recall and precision values are given for each of the three methods and for each class. The mean recall and precision, and the overall accuracy, are given at the bottom of the table.

The Morphological descriptors give good performances, with 83.6% accuracy (51.7% improvement over Local Trend and 18.1% over Global Trend, see Tab. 1).

It is interesting to look at the results class-by-class, also comparing with sec. 2.5.3. The Morphological descriptors have been designed to embed the main characteristics of the categories into a compact and effective representation. The overall conclusion which can be drawn is that Morphological descriptors have a better performance because they work rather well on all categories, giving comparable recall/precision. This is not the case for Local and Global Trend, which have instead one or more categories with particularly bad results.

Analysing the confusion matrix of the recognition by the Morphological descriptors, in Tab. 3, it can be pointed out the improved discrimination between `down` and `up/down` compared to the previous methods. However, the matrix shows that the issue is still present. Similarly there is confusion between `up` and `down`.

[2]Pitch and spectral centroid are used alternatively, choosing the former only when Φ_6 (median pitch strength) is above 0.4.

Methods	Abstract Morphol.	
Measures	Rec.	Prec.
up	87.7	79.6
down	71.5	73.7
up/down	76.3	76.2
impulse	91.5	91.9
repetition	90.3	93.2
stable	85.8	92.4
Average	83.9	84.5
Avg. Accuracy	**83.6**	

Table 2: Recognition results of the **Abstract** family using the AbsMorpho descriptors (averaged over 5 crossvalidation folds).

Categories	1	2	3	4	5	6
up - 1	**292**	20	13	2	2	4
down - 2	32	**244**	49	9	6	1
up/down - 3	15	47	**273**	11	6	6
impulse - 4	4	11	8	**301**	3	2
repetition - 5	9	7	6	5	**300**	5
stable - 6	16	3	11	0	5	**213**

Table 3: Confusion matrix for the **Abstract** family using the AbsMorpho descriptors (corresponding to results in Tab. 2).

`repetition` is well recognized, thanks to the presence of the specific Ψ_3 descriptor. Both `impulse` and `stable` are confused, even if not to a large extent, with `up`, `down` and `up/down`. This could be explained by the fact that the system identifies rising or falling cues even in the `impulse` and `stable` imitations, and provides a classification according to this.

Recognition of the all families using the GenMorpho descriptors.

The set of GenMorpho descriptors introduced in sec. 2.6.2 has been used to do automatic category recognition on the three SkAT-VG dataset families. In this section are presented the recognition results obtained so far.

The SkAT-VG dataset is divided in two parts: Voice Only (VO) and Voice plus Gesture (VG) recordings/sessions. Moreover, the subjects are allowed to use up to 5 trials to improve their imitations. We have begun the development of the classifiers by using only data from the VO experiments, and taking only the last trial of each subject (assuming this should be the one with higher quality). In a following phase we have moved toward the use of the whole dataset, taking data from both VO and VG and from all the trials.

The results have been 10-fold crossvalidated, avoiding the presence of recordings from the same subject in both train and test set at the same time. All the presented figures have been obtained after an optimization of two SVMs parameters (by grid search and subcrossvalida-

Dataset	VO/LastTrial		All Data	
Categories	Recall	Precision	Recall	Precision
up	83.78	75.92	82.15	78.73
down	73.11	71.72	77.76	73.89
updown	69.28	80.37	70.06	74.59
impulse	90.89	93.48	91.51	92.04
repetition	91.89	98.00	93.15	96.24
stable	95.78	93.89	96.13	97.34
Mean val.	84.12	85.56	85.12	85.47
Accuracy	83.97		84.35	

Table 4: Recognition results of the **Abstract** family using the GenMorpho descriptors (averaged over the 10 crossvalidation folds).

tion): σ, which controls the sharpness of the gaussian kernels, and C, which manages the soft margin of the classifiers.

Dataset	VO/LastTrial		All Data	
Categories	Recall	Precision	Recall	Precision
blowing	72.00	73.71	68.41	72.33
whipping	72.44	73.93	79.74	81.46
shooting	68.06	74.75	69.42	75.32
crumpling	66.67	68.09	68.22	68.43
rolling	67.72	64.96	63.25	65.57
rubbing	61.28	66.54	67.41	70.78
hitting	81.44	76.78	82.90	77.90
dripping	88.31	79.79	91.11	82.39
filling	49.19	60.44	54.20	57.82
gushing	74.56	71.73	72.26	71.94
Mean val.	70.17	71.07	71.69	72.39
Accuracy	70.44		71.58	

Table 5: Recognition results of the **Interaction** family using the GenMorpho descriptors (averaged over the 10 crossvalidation folds).

After a large set of early experiments, we have found that classification results are improved using optimal subsets of descriptors, different for each family.

The Abstract family has best results using all the descriptors; as reported in Tab. 4, accuracy is about 84% on the reduced dataset, and rises to 84.3% using all the data.

For the Interaction family we have obtained the best results by excluding the descriptor Φ_{10}, that is the standard deviation of the spectral centroid. Number figures are reported in Tab. 5. Accuracy is around 70.4% on the data subset, and slightly increases to 71.6% on the full collection of recordings.

Dataset	VO/LastTrial		All Data	
Categories	Recall	Precision	Recall	Precision
alarms	67.16	84.99	65.65	78.50
buttons	89.25	82.86	88.09	84.89
doors	84.25	88.93	88.32	88.30
filing	87.75	85.66	87.00	85.90
fridge	40.03	47.35	46.52	55.94
mixers	59.01	45.22	66.07	48.07
printerfax	54.86	61.48	52.28	62.66
windshield	50.25	57.91	52.18	56.44
vehicleext	65.56	67.94	71.40	72.98
vehicleint	62.02	56.03	63.97	65.29
Mean val.	66.01	67.84	68.15	69.90
Accuracy	66.55		69.06	

Table 6: Recognition results of the **Machine** family using the GenMorpho descriptors (averaged over the 10 crossvalidation folds).

About the Machine family, best results have been provided by not using the descriptor Φ_7, that is the median of the pitch. The Tab. 6 shows the results, which are slightly worst than with the Interaction family. Accuracy is about 66.5% on the reduced dataset, and rises to 69% on the full dataset. We point out that in this case, as for Abstract family, a larger amount of data improves results; for Interaction family the opposite is true.

2.7 Recognition using time series comparison through DTW

In sec. 2.5 and 2.6 various methodologies have been proposed to model the time series of the low-level features. Good results on blind classification have been obtained on all the three families of the dataset.

Despite this, the lower results on Interaction and Machine families (sec. 2.6.3) prove that time series modeling is not trivial for these. We then have focused on the direct exploitation of the series themselves. Our first step to do this has been the definition of a distance measure between time-aligned series, such as the Dynamic Time Warping (DTW) (sec. 2.7.1). Using this distance it has been possible to study automatic classification strategies which use or do not use clustering (sec. 2.7.4 and 2.7.3 respectively). Finally, classification performances results are shown in sec. 2.7.7.

2.7.1 Time aligned distance by DTW algorithm

We introduce here the DTW algorithm which will be used in the following sections. The procedure is defined on time series of scalars, thus series of vectors are excluded. For two given time series $T_1(t)$ and $T_2(t)$ of length t_1 and t_2 respectively, a matrix M is computed. It has size $[t_1, t_2]$ and is initialized to 0 on the first column ($\forall i = 0$) and on the first row

($\forall j = 0$). When $i \neq 0$ and $j \neq 0$ it is computed as:

$$M(i,j) = k[T_1(i) - T_2(j)]^2 + \min\{M(i-1,j-1), M(i-1,j), M(i,j-1)\} \quad (7)$$

where

$$k = \begin{cases} 1 & \text{if } \operatorname{argmin}\{M(i-1,j-1), M(i-1,j), M(i,j-1)\} = 1; \\ 1.5 & \text{otherwise.} \end{cases} \quad (8)$$

The matrix M is thus weighted in such a way to privilege the alignments without lags. The final distance value between T_1 and T_2 is found in the last position of M:

$$d_{DTW}(T_1, T_2) = M(t_1, t_2). \quad (9)$$

We have worked on the effectiveness of the DTW by developing a preprocessing for the time series (sec. 2.7.2). Our first approach have been based on DTW distances, without any kind of optimization (sec. 2.7.3). Some results from the analysis on the dataset, exposed in D4.4.1, give a strong evidence that for each reference stimuli, the subjects can use different imitation strategies; we have thus applied a per-reference sound clustering before the actual classification (sec. 2.7.4). This has also the advantage of reducing the computational complexity, but we finally found that better results are obtained without using clustering at all. A final resume of our results is then presented in sec. 2.7.7.

2.7.2 Time series preprocessings

Having decided to use DTW as distance measure, we need to develop an effective representation of the time evolution of the audio signal. We are interested into a set of descriptors with some relevant characteristics:

- it has to capture the main cues of the signal, both in time and frequency;
- signals can be both harmonic and/or inharmonic, and this has to be quantified;
- we are interested in the relative variations of the signal, hence a degree of invariance to absolute values is desirable;
- despite previous point, also absolute values can bring relevant information and should be used.

The development of descriptors reflects these needs, and started by selecting some relevant low level features $F_i(t)$ (described in sec. 2.3): Loudness, Spectral Centroid, Spectral Spread, Zero crossing rate, LPC-min. Each sound in the dataset is associated with its set of time series $F^a = \{F_i(t) \forall i \in [1, \ldots, 5]\}$, which undergo the following processing:

1. Each $F_i(t)$ is normalized to have zero mean and unit standard deviation, obtaining $\bar{F}_i(t)$.
2. A new set of features F^b is obtained joining F^a and $\bar{F}$.
3. Each $F^b_i(t)$ is smoothed with a linear phase low pass filter.

4. Each $F_i^b(t)$ is differentiated, obtaining an approximation of its derivative $F_i^{b'}(t)$.
5. A new set of features F^c is obtained joining F^b and $F^{b'}$.
6. In each $F_i^c(t)$ are usually present head and tail silent portions; these are deleted exploiting knowledge of Active Regions boundaries (see sec. 2.3).
7. Each $F_i^c(t)$ is then decimated by a factor 4, leading to $F_i^d(t)$; this reduces time series length and eases the following DTW computations.
8. The final set of descriptors D is exactly F^d; a second set of descriptors D' is also found, resampling each $F_i^d(t)$ to a specific length, usually 250 samples.

The set D' has been used only in studies of intra-category clustering (sec. 2.7.4).

From now on, we will indicate with $D(s)$ the set of all the descriptors D issued from a specified sound s, and with $D'(s)$ the D' descriptors from the same sound s.

2.7.3 No optimization

Fixed length series The audio dataset has been analysed by means of the GenMorpho Descriptors (sec. 2.6.2). Analysis in Task 4.3 of WP4 (see D4.4.1) confirmed that among all the imitations of a single reference sound, the subjects use different strategies. The vocal recordings belonging to a given category can thus be very different from each other. During train phase, all the train set descriptors from the set D' (resampled to fixed length) are stored.

The test phase is based on a k-NN search done separately for each descriptor; DTW distances are computed between each test sound s and all the train set instances. The output category label is decided as the mode of the labels found by each descriptor.

Variable length series Results on set D', that is with resampled fixed-length time series, show a confusion between categories which have very different mean time durations. This suggests that using set D' while avoiding clustering lead to a loss of information. We thus moved to the non-resampled set D, in which each serie has its original time length. The train and test procedure remains the same as before. Results are clearly improved on all the three families, as shown in sec. 2.7.7.

2.7.4 DTW optimization using intra-category clustering

In order to lower the computational complexity of DTW search, we model each set of reference sound imitations by means of a clustering, which should in principle highlight the different imitation strategies and better modelize them.

The time series of descriptors D/D' are defined on different spaces, because of their different meanings; defining a common distance measure over them is not straightforward. The clustering described in the following is thus applied separately to each descriptor. Only during the classification phase, using a *late fusion* approach, the results of the clusterings are merged.

It has been chosen to use Dynamic Time Warping (DTW) as distance measure between time series (sec. 2.7.1). Despite the use of DTW, the sequence length has a direct effect on

the distance values, thus inducing a clustering between similar-length series. To avoid this effect, D' has been chosen instead of D, hence having all the descriptors series resampled to a given number of points (time duration).

The distances d^i_{DTW} are found using a DTW algorithm applied to the i-th time serie of the descriptors set D'. For each reference sound, a clustering is computed during the training phase. This is achieved by finding $d^i_{DTW}(s_a, s_b)$ between each pair of recordings s_a and s_b in a given set of recordings of the same reference sound. The set of $d^i_{DTW}(s_a, s_b)$ is then used to obtain $cl = 5$ clusters by means of a standard clustering algorithm. Each reference sound is modeled by cl time series, obtained as the time mean of all the series associated with a given cluster. The train set is thus modeled in this way: for each descriptor, we have cl mean time series to represent each of the 20 (or 12) reference sounds.

In the test phase it is given a sound s with its own descriptor set $D'(s)$. For each time serie in $D'(s)$, its DTW distances are computed against each mean time serie from the train phase. A reference sound label is decided from each descriptor: it is taken as the label of the cluster with contains the nearest time serie. Using late fusion, the category label for s is decided as the mode of the per-descriptor labels, mapping reference sounds back to their categories.

Several different values of cl have been tested, and recognition performances improve steadily while increasing cl; this suggests that an approach based on clustering could be not optimal, hence we move toward a classification strategy which does not rely on clustering at all.

2.7.5 Use of kNN

During the test phase, if we do not rely on clustering optimization, we obtain the DTW distances of the given recording s from *all* the train series. With the approach described so far, the label of the nearest instance is used to do classification. This procedure is widely known as k-Nearest Neighbor (kNN) algorithm, having $k = 1$. An immediate extension is to rise k to some other value. Test phase of classification is thus this:

1. for a given sound s, all the DTW distances between its descriptors and the stored ones are computed;
2. for each descriptor of s, decide a label by looking at the most frequent label among the k nearest points in the train set;
3. classify s by the label which is more often found at the previous step.

We have chosen to use a value of $k = 10$. This gives better results almost always; only an experiment on Abstract family gave worst recognition performances.

2.7.6 Refinements: Feature selection and k optimization

After a set of experiments, we found a configuration for the recognition system which gives reasonably good results:

- Use of non-resampled descriptors (set D);
- No optimization by clustering;

- Use of k-NN algorithm for recognition, with $k = 10$.

In sec. 2.7.7 (Table. 10) are shown the results obtained with this configuration, applied on the three families.

Having established the details of a working system, we move toward the tuning of it. As a first step, the set of low-level audio features listed in sec. 2.7.2 has been enriched; we include: Inharmonicity, Loudness, Total Energy, Noisiness, Spectral Centroid, Spectral Rolloff, Spectral Spread, Zero Crossings, $1st$ Formant (LPC-min), $1st$ Formant Bandwidth, Spectral-peak-min, Pitch, Pitch Strength. We also use a more fine-grained definition of the descriptors set D. While the computational procedure remains the same, we now consider separately its four distinct modelings:

- D_1, smoothed (low pass filter) original input descriptors;
- D_2, smoothed and normalized input descriptors;
- D_3, differentiated D_1;
- D_4, differentiated D_2.

We then apply the following feature selection procedure, based on a bidimensional grid-search. The 13 basic time series are arranged in 14 possible combinations: a first one in which all the series are considered, and then 13 leave-one-out combinations. Similarly, the 4 modelings D_m are selected according to all their possible (binary) $2^4 = 16$ combinations (minus the "0000" one which is meaningless: it takes no modelings at all). Hencefore we have a grid for all the possible choices of descriptors and modelings: we explore it by running a 10-fold crossvalidation for each case, obtaining a matrix A of size 14×15 with all the found accuracies. We then compute an average by the rows of A, and obtain a column vector with the average accuracies *per-modeling*. We take the higher scoring modeling as the best one for a given dataset family:

- Abstract: $D^*_A = \{D_2, D_3\}$;
- Interaction: $D^*_I = \{D_1, D_2, D_4\}$;
- Machine: $D^*_M = \{D_1, D_2, D_4\}$.

The selection of descriptors is done in a slightly different way, because testing all the possible subsets is not feasible. Considering the matrix A, we do an averaging by the columns, thus obtaining a row vector with the mean accuracies given by each leave-one-out combination. We then sort this vector in increasing order, along with its associated descriptors list: being the poorest accuracy the first one, the corresponding leaved-out descriptor is the one which has the worst impact on performance when *not* used. In other words, we sort the descriptors from the most effective to the least one in a vector d of length 13. We then do a linear grid search along d, keeping fixed the modelings to the D^*_x set of each family. At each step k of the search, we use the first k components of d and do a 10-fold crossvalidation. Looking at the output accuracies we found, as expected, a bell-like shape: at the beginning we have too few (effective) descriptors, then the set is large enough to have good performances, and finally we add the *bad* descriptors which actually makes performances worse. The optimal sets of descriptors found for each family are:

- Abstract: Loudness, Spectral-peak-min, Total Energy, Pitch, Inharmonicity, LPC-min, Spectral Centroid;
- Interaction: Loudness, Total Energy, Zero Crossings, LPC-min, Pitch Strength, Inharmonicity, Spectral Centroid;
- Machine: Loudness, Total Energy, Pitch Strength, Pitch, Noisiness, Spectral Spread, Spectral-peak-min.

These sets, along with the modelings chosen above, have been used to finally tune the value of k in kNN algorithm. Another linear search have been done, for each family, which led to the following best values: 3 for Abstract, 5 or 8 of Interaction, 3 for Machine. Final results shown in sec. 2.7.7 have been obtained with these configurations.

2.7.7 Results on the three families

In this section we present the results which have been obtained from our experiment on the SkAT-VG dataset. In a first part are reported some early results which have driven the development of the classifiers, as described in the previous sections. Then, we show the recognition results on all the three dataset families, using the descriptors and the classification strategies exposed in sec. 2.7.3.

Early results These first experiments have been done almost only on the Abstract family, because it is the easier to classify: a comparison can be made in sec. 2.6.3. The number figures have been obtained using a slightly different descriptors set than D/D': in this case it is not present the differentiation of the time series. While these results are not comparable with the main ones, exposed in the following, we think it is interesting to report them as they support the decisions taken along the development of the classifiers.

A first attempt to the automatic classification of the Abstract family has been without the use of clustering, and is shown in Tab. 7. The used descriptors are very similar to D', because they are resampled to have a fixed length of 250 samples, but they lack the differentiation in time.

We have then tried to optimize the search by using the clustering, but results are worst, as shown in Tab. 8. We have found that rising the number of clusters improved the results, thus we conclude that the clustering optimization is not working well.

It has been noticed that the confusion matrices associated with both Tab. 7 and 8 (not reported) show that `down` and `impulse` are confused. This has been taken as an indication that the resampling to an identical length of the time series is not optimal. We have thus applied the same procedure as above, but using non-resampled time series; this eventually implies a clustering between series with similar lengths, but classification is not adversely affected, as shown in Tab. 9.

We have finally decided to avoid the use of clustering and to rely on non-resampled descriptors; with minor modifications the descriptor set D has been developed (already described previously) and a first complete group of results is shown in Table 10. The use of kNN with $k > 1$ usually improves results.

This constitutes the base for the refinements exposed in sec. 2.7.6, which have been used for the following main results.

Measures	Up	Down	Updown	Impulse	Repetition	Stable	**Mean**
Precision	70.351	58.744	66.033	82.398	97.095	93.814	78.073
Recall	88.737	50.579	66.848	81.842	94.947	76.667	76.603
Accuracy	76.704						

Table 7: Recognition results of the **Abstract** family using DTW without optimization, averaged over the 5 crossvalidation folds.

Measures	Up	Down	Updown	Impulse	Repetition	Stable	**Mean**
Precision	54.422	39.501	54.006	71.335	82.940	76.437	63.107
Recall	68.316	35.000	47.146	65.737	97.000	64.444	62.941
Accuracy	63.101						

Table 8: Recognition results of the **Abstract** family using DTW with optimization (clustering on early descriptors), averaged over the 5 crossvalidation folds.

Measures	Up	Down	Updown	Impulse	Repetition	Stable	**Mean**
Precision	71.905	68.235	68.506	84.482	94.209	98.462	80.966
Recall	87.842	62.105	73.953	92.947	93.895	62.222	78.827
Accuracy	79.163						

Table 9: Recognition results of the **Abstract** family using DTW using non-resampled early descriptors, averaged over the 5 crossvalidation folds.

	Dataset	VO/LastTrial				All Data			
	k	10		1		10		1	
Family	Measures	Rec.	Prec.	Rec.	Prec.	Rec.	Prec.	Rec.	Prec.
Abstract	Mean val.	86.35	88.00	88.65	89.87	85.84	87.04	85.90	86.81
	Accuracy	86.41		**88.67**		85.69		85.58	
Interaction	Mean val.	69.45	71.29	66.64	69.33	67.05	70.04	64.99	66.42
	Accuracy	**69.57**		66.78		67.40		65.41	
Machine	Mean val.	64.02	65.91	61.17	61.30	65.75	70.09	63.58	66.18
	Accuracy	64.56		61.73		**67.00**		64.91	

Table 10: Recognition results of the three families using DTW on descriptor set D and K-NN (with k=1 and k=10), averaged over the 10 crossvalidation folds.

Main results The results are presented separately for each family, because the classifiers are well distinct. Similarly to sec. 2.6.3, we present two groups of results: one obtained using only the last recording trials from the Voice Only sessions (VO/LastTrial condition), and a second

found on the whole dataset (recordings from Voice Only and Voice plus Gesture sessions, with all the trials).

In each table are shown the results using the kNN algorithm with optimum values for each family and dataset. All the results are 10-fold crossvalidated, taking care of not having the same subject in both train and test set.

We begin by presenting our results on the Abstract family, reported in Tab. 11. The results are better using only the reduced dataset, and in this case the value $k = 3$ is the best one, with 92.64% accuracy. The complete dataset provides about 91.66% accuracy, with the same value for k. Given the structure of the dataset, in which every category is defined using two different sound stimuli, we present also the confusion matrix which is obtained doing the recognition by reference sounds instead of by categories (Fig. 8). It can be seen that, apart from the sound `updown2`, the three categories `up`, `down` and `updown` are the most difficult to recognize. Also, the imitations of `stable2` are not well recognized.

For the Interaction family the best results are obtained with different values of k for the two recognition experiments: 5 for VO/LastTrial and 8 for AllData (Tab. 12). The best achieved accuracy, using data from the reduced dataset, is 72.12%. Despite this rather good measure, paired with a mean recall of 71.83%, it should be noticed that on some categories the performances are very poor. In particular, recall reaches only 21% for `filling` and about 55.8% for `rolling`. Both these categories have the worst performances on all testing conditions. These difficulties are confirmed by the confusion matrix (Fig. 9), where it is clear that `filling*` sounds are confused with a large group of others, especially from `crumpling1` and `rolling*`.

The Machine family is the least easily recognizable, as already noticed in sec. 2.6.3, and the only one which has a performance gain when exploiting all the available data. Numerical figures are reported in Tab. 13: using the whole dataset we achieve an accuracy of 71.83%. This has been obtained with $k = 3$, which always provides the best performances on this family. As for Interaction, also in this case there is one category performing particularly bad, that is `printerfax`; also `windshield` is not well recognized. It is interesting to point out that both categories, especially `printerfax`, are defined using reference sounds which are composed and complex. This inevitably rises the intra-category variability, thus making recognition tougher. Once again the confusion matrix (Fig. 10) supports these findings, also showing an unexpected misclassification between `windshield*` and `alarms2`.

2.8 User relevance feedback

So for we have discussed the automatic recognition of imitation categories within the SkAT-VG dataset. In sections 2.5, 2.6 and 2.7 we use a blind classification approach to tackle this problem. The SkAT-VG project has however a broader scope than the classification, as it is focused on the interaction with the user. This has an important meaning, as the evaluation of our classifiers changes dramatically: an optimal recognition is not the one which found the right class, but the one which drives the user nearer to a desired sound. This opens our research toward a context in which the user can, and should, interact which the classifier; there is thus the need to introduce some form of *user feedback* in the classification procedure.

In sec. 2.8.1 we introduce a system which exploits the user feedback to enhance the browsing of a sound dataset in the context of a *search by similarity*. We then discuss in

Dataset	VO/LastTrial		AllData	
Measures	Rec.	Prec.	Rec.	Prec.
Up	95.89	92.25	94.21	92.84
Down	80.33	93.76	82.94	91.10
Updown	93.00	88.07	90.71	85.02
Impulse	96.89	90.38	96.85	89.18
Repetition	97.00	96.36	95.22	96.70
Stable	92.17	100.00	90.66	99.55
Mean val.	92.55	93.47	91.77	92.40
Accuracy	**92.64**		91.66	

Table 11: Best recognition results of the **Abstract** family: DTW on descriptor set D_A^* and KNN (with $k = 3$), averaged over the 10 crossvalidation folds.

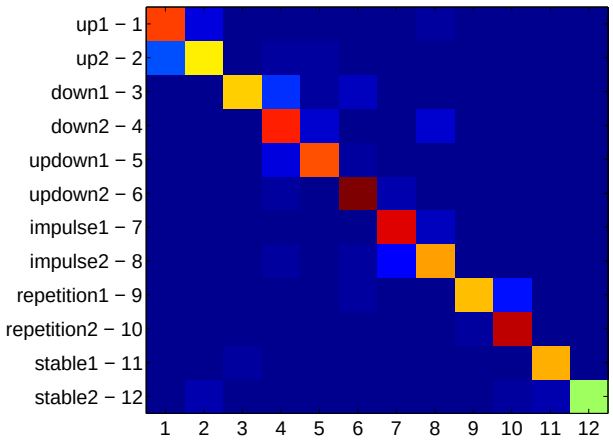


Figure 8: Confusion matrix for the reference sounds of the **Abstract** family using DTW on descriptor set D_A^* and KNN (with $k = 3$), averaged over the 10 crossvalidation folds.

sec. 2.8.2 a possible porting of this kind of user feedback toward our classification procedures based on DTW (sec. 2.7.2).

2.8.1 Synthassist

Cartwright et al. introduced a system called Synthassist [CP14]. It is a software prototype conceived to ease the browsing of a sound dataset, which has been collected by sampling the parameters space of a synthesizer. It contains about 10000 recordings from the audio output of a sound synthesizer, which has been driven using sparse values along its whole parameters space. We have implemented from scratch the Synthassist prototype, and we have paired it with an automatic evaluation procedure.

Dataset	VO/LastTrial		AllData	
Measures	Rec.	Prec.	Rec.	Prec.
Blowing	68.11	70.50	61.99	76.11
Whipping	92.00	69.48	92.78	72.30
Shooting	71.44	73.81	69.83	75.86
Crumpling	76.33	60.24	78.03	59.11
Rolling	55.83	61.82	54.32	56.21
Rubbing	71.67	85.29	75.43	83.84
Hitting	86.89	80.23	88.76	77.64
Dripping	91.64	89.05	89.74	84.87
Filling	21.22	56.67	22.18	85.85
Gushing	83.11	75.07	78.97	70.47
Mean val.	71.83	72.22	71.20	74.22
Accuracy	**72.12**		71.55	

Table 12: Best recognition results of the **Interaction** family: DTW on descriptor set D_I^* and KNN (with $k = 5$ for VO/LastTrial and $k = 8$ for AllData), averaged over the 10 crossvalidation folds.

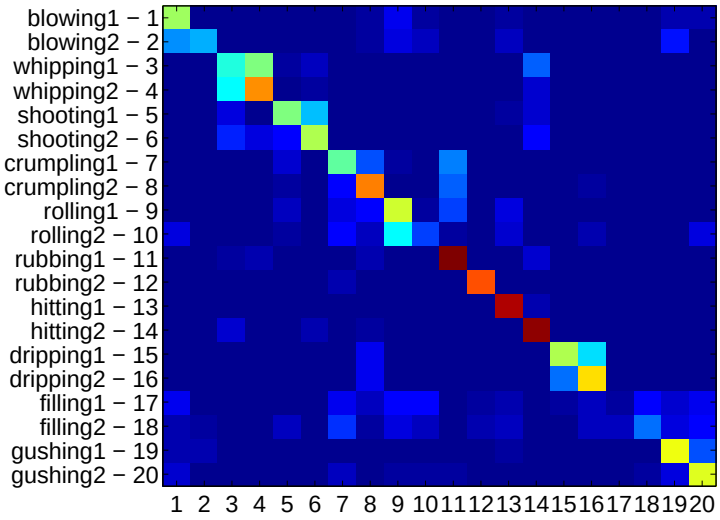


Figure 9: Confusion matrix for the reference sounds of the **Interaction** family using DTW on descriptor set D_I^* and K-NN (with $k = 8$), averaged over the 10 crossvalidation folds.

Synthassist workflow The authors of the Synthassist system proposed the following workflow:

- The user give to the system one or more audio examples, possibly recording his/her own voice, as *query*;

Dataset	VO/LastTrial		All Data	
Measures	Rec.	Prec.	Rec.	Prec.
Alarms	84.53	63.55	86.74	59.89
Buttons	90.50	88.40	90.08	88.44
Doors	93.00	83.55	93.14	80.49
Filing	90.25	74.11	90.61	76.96
Fridge	47.07	79.67	55.40	66.16
Mixers	77.71	55.43	69.21	51.04
Printerfax	28.58	88.00	30.81	78.87
Windshield	50.00	65.37	41.17	72.73
Vehicleext	92.50	75.47	90.57	78.59
Vehicleint	54.81	75.95	60.12	82.17
Mean val.	70.90	74.95	70.78	73.53
Accuracy	71.43		**71.83**	

Table 13: Best recognition results of the **Machine** family: DTW on descriptor set D^*_M and KNN (with $k = 3$), averaged over the 10 crossvalidation folds.

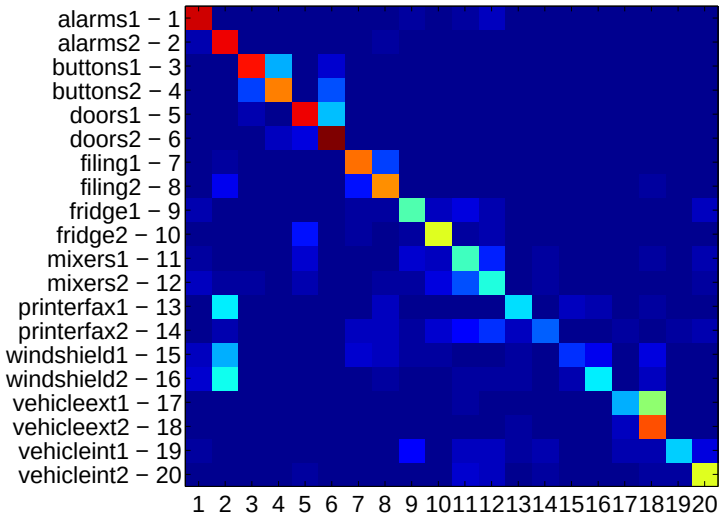


Figure 10: Confusion matrix for the reference sounds of the **Machine** family using DTW on descriptor set D and KNN (with $k = 3$), averaged over the 10 crossvalidation folds.

- The system extract some audio features from the input sound(s), and use them as a search key through the dataset;
- The user is provided with two output sounds sets: one with possible results, and to other to be evaluated;
- The user evaluates the sounds according to his/her target, and evaluated sounds are

added to the original query;

- The system changes its search strategy according to the feedback, by using different weights for the features;

- A new set of results is proposed, thus iterating the procedure until the user is satisfied.

Let's see in more detail the search and update steps of the procedure. We will then present our evaluation experiments.

Automatic search and ranking The search by similarity, through the whole dataset, starts by extracting some low level audio features from the input files. These are [Pee04]: pitch, loudness, inharmonicity, spectral centroid, spectral spread, spectral kurtosis and clarity. The latter is a measure of how much a sound is "coherent" [MW05]. These features are then augmented by their standardized versions, which have zero mean and unitary variance.

Hence, to describe an input sound s are used 14 time series, which are stored in the matrix $\mathbf{X}_s = \{\mathbf{x}_1, \ldots, \mathbf{x}_{14}\}$. The same set of features is extracted and stored from the whole sound collection in the database.

Between each pair of sounds s_i and s_j a distance measure $d_{DTW}(s_i, s_j)$ is defined over this set of features:

$$d_{DTW}(s_i, s_j) = \sum_{d=1}^{14} w_d \mathrm{DTW}(\mathbf{x}_d^i, \mathbf{x}_d^j), \tag{10}$$

where $\mathrm{DTW}(\mathbf{x}_d^i, \mathbf{x}_d^j)$ is found by the Dynamic Time Warping algorithm, already discussed in sec. 2.7.1, applied on the d-th feature time serie. The scalar values w_d are called *weights* and are initialized to be all equal to each other; user feedback is then used to update them (see below).

The search through the database is simply accomplished by a ranking, which presents to the user the nearest k sounds found in the database according to the distance d_{DTW}. As anticipated, the user is asked to evaluate a second set Z of sounds; this is built by taking the single nearest sound to the input query according to each feature[3].

User feedback The user feedback has two effects on the search by similarity.

Updating query The user input query is updated to encompass also the relevant sounds from the set Z. The Synthassist authors make use of a technique called Prioritized Average Shaping (PAS) [NR], which computes a *mean time serie* among several series. The PAS method is applied to each feature separately, and begins by performing an agglomerative clustering on the series, based once again on the DTW distance. Then, with a bottom-up approach, series are time-aligned to each other and averaged; this latter averaging is in turn based on the user-provided relevance scores. The PAS procedure ends up by providing a new query representation $\bar{\mathbf{X}}$, which is then used to do the following searches in the dataset.

[3] Actually, only half of these sounds are put in Z: a random selection among the 14.

Updating weights The weights of the features w_d are updated according to the following formula:

$$w_d = \left[\frac{1}{\sum_{k=1}^{|Z|} s_k} \sum_{k=1}^{|Z|} s_k \mathsf{DTW}(\mathbf{y}_d^k, \bar{\mathbf{x}}_d)^2 \right]^{-\frac{1}{2}} \tag{11}$$

where: s_k are the relevance scores given by the user to the sounds in the Z set, $\mathbf{y}_d^k$ is the d-th feature of the k-th sound in Z, and $\bar{\mathbf{x}}_d$ is the d-th feature of the updated input query sequence.

Our evaluation In order to understand the effectiveness of Synthassist in finding sounds which are actually similar to a given one, we exploit the ground-truth knowledge about sound categories into a dataset. Out evaluation procedure, coupled with the Synthassist one, is the following:

- A query sound is selected from the dataset, and excluded from it; the query is known of being from category C.
- The main Synthassist routine is called on the selected query, storing the topmost 10 results in R^1.
- Synthassist asks to rate sounds; our automatic rating exploits dataset ground-truth knowledge, giving rating 1 to each sound from the same category C of the query, and 0 to sounds from other categories. No half-way score is given, thus assuming an high degree of intra-category similarity and inter-category dissimilarity.
- The automatic rates are provided to Synthassist, thus weights and query are updated. Process is repeated to find R^2 and R^3.

The original dataset is not available, hence our evaluation has been done using another dataset: the UrbanSound8K [SJB14]. We used only the *salient* sounds from the dataset, hence having 5702 sounds on 10 categories. For each of the ten categories, we randomly choose, once for all, 15 examples (so we use 150 sounds as query out of the 5702).

The effectiveness of Synthassist is evaluated by looking at how many sounds from the same category C as the query are found. The Tab. 14 presents the numerical figures found by our analysis. We compare three different setups: the `Baseline` system, as described so far, and two alternative versions of it: updating either only the query (`Q only`) or the weights (`W only`). In Tab. 14 are reported the numbers of correctly found sounds among the first three iterations of search, by looking only at the topmost k results, with $k \in [3, 5, 10]$. Optimal results would be to have in all cases a value equal to k.

Some conclusions can be drawn. First, the baseline system is usually the best, hence supporting the need to update both the query and the weights. The UrbanSound8K dataset is rather tough for classification, and Synthassist performs quite well: in each experiment almost half of the results belong to the same category of the query; this could be see as an accuracy of about 50%: a rather good figure considering the strong assumptions about inter-/intra-category (dis)similarity.

Topmost k	Update rule	It. 1	It. 2	It. 3
3	Baseline	1.28	1.73	1.71
	W only	1.28	1.42	1.45
	Q only	1.28	1.43	1.45
5	Baseline	2.09	2.67	2.63
	W only	2.09	2.32	2.33
	Q only	2.09	2.19	2.19
10	Baseline	3.81	4.68	4.61
	W only	3.81	4.49	4.60
	Q only	3.81	4.01	3.99

Table 14: Evaluation of Synthassist performances on UrbanSound8K dataset.

Despite this, we can see that the third iteration for Baseline is usually worst than the second, henceforth the improvement is not monotonic. Another interesting point is that, at the third iteration, the weights-only update is almost as effective as the baseline.

2.8.2 Portability toward our approach

Synthassist has several important aspects related to the use of relevance feedback by the user. First, the system is based on a distance defined on the feature time series, thus exploiting them directly without any modeling. Henceforth, the system can update the search query according to the user input: the immediate availability of the time series is easily exploited by averaging between them. Feature weighting is also straightforward, because each feature is treated by itself.

The sec. 2.7 describes our approach to the automatic category recognition by using time series. This method has slightly lower performances compared to the GenMorpho descriptors (sec. 2.6.3 and 2.7.7), but is still satisfactory and can be exploited. Similarly to Synthassist, our method uses a distance based on DTW algorithm to compare the feature series.

We can thus envisage a use case for our classification system in which the user can interact to drive classification toward a desired target. This is strongly inspired from Synthassist, but in out case we are interested into category labels.

In order to do this, our system has to be modified to provide a *rank* instead of a hard classification; this can be easily accomplished since distance measures are available for all the categories, enabling a full ranking. For a given input sound, the system should provide a list of possible categories for it, along with some compelling examples selected from each class. Following the workflow of Synthassist, the user should listen and rate some of the examples. Our modified system can then make use of the relevance feedback in two ways.

The first one, straightforward, is to repeat the search on the dataset by using the feature weighting given by a slightly modified version of equation 11 from sec. 2.8.1. Moreover, the sound examples which have been rated by the user can be averaged with the original input data; this can be accomplished by using the same *early* averaging of Synthassist, of by other means based on *late* distance/score averaging. This leads to a system very similar to Synthassist, but which is built on category labels.

2.9 Summary and discussions

2.9.1 Results and novel contributions

In this section we have described three different methodologies to achieve blind classification of imitations using only the audio signal.

Our first approach is based on time modeling by HMMs, and we have developed two novel sets of descriptors: one based on local signal variations (Local Trend), and the other on long-term variations (Global Trend). We have applied these strategies to the Abstract family only, obtaining accuracy of 55.1% and 70.8% respectively.

After this, we have moved to a novel approach. We have developed new sets of descriptors which are tailored to find and measure the acoustic cues of the imitation signals. A first set of morphological descriptors has been developed for the Abstract family only (AbsMorpho), obtaining an accuracy of 83.6%. An international conference article on these results has been published [MP]. The same approach has been applied to the other two families. The best accuracies, obtained with a new set of morphological descriptors (GenMorpho), are respectively: 84.3% for Abstract, 71.6% for Interaction and 69.0% for Machine.

We point out that the set of GenMorpho descriptors has also been used in WP4, leading to interesting findings in Task 4.3.

Our last step in the development of a blind classifier for the vocal imitations has been toward the direct use of low level features. After a phase of early studies, we have found a straightforward methodology, based on DTW distance, which achieves good classification performances. Despite its relatively simple formulation, the approach is of particular interest because works on the three families without per-family tunings. The best accuracies with this method are: 91.66% on Abstract, 71.55% on Interaction and 71.83% on Machine.

In the context of SkAT-VG project, our last approach is particularly interesting. In fact, several use scenarios have been conceived within the SkAT-VG project and, in most of these, it is advisable for the classification system to be able to exploit a user relevance feedback. With this purpose, we have assessed the effectiveness of the Synthassist system, proposed by Cartwright et al. [CP14]. We have verified that the main ideas behind Synthassist can easily be ported to our last recognition system (the one based on direct time series use); we can therefore conclude this latter methodology has the further advantage of a relatively straightforward relevance feedback integration.

2.9.2 Future steps

The future work which has to be done in WP5, related to audio, can be divided in two distinct parts: 1. refinements to be applied to our recognition methodologies and 2. tasks for the integration of our classifiers into SkAT-VG.

For the first part, being the DTW-based system the one with best performances, our commitment will be to advance its development. The recall for some categories is very poor: this is not completely justified and need to be studied.

The blind recognition based on Morphological descriptors is not giving the best performances, but has still an advantage over the direct use of time series: it scales better if applied on bigger datasets, because the classification is less computationally demanding. We thus will

continue the development of this methodology too, at first by validating the choice of kernel parameters for the SVM classifiers.

The integration of our classifiers into the SkAT-VG prototypes is the next step in the context of the project. From a technical point of view, we need to port our classifiers (both the signals processing and the statistical models) into the real-time Max software environment. Max is the reference software platform for the whole project, hence it is also used for gesture recognition (see Fig. 24 for example). Once the porting will be done, the shared environment should ease the development of multimodal recognition, combining gesture and voice. A separate task for the integration of our work into the project use cases will be the integration of relevance feedback into classification, as anticipated in sec. 2.8.1.

3 Gestures analysis and recognition

The choice of the movement analysis method on the SkAT-VG imitation dataset was driven by the insights gathered through the qualitative studies on participants' strategies, as reported in D4.4.1. In particular, the participants' movement were found, as expected, from the use of different body part (fingers, hand, head, trunk) to the movement direction (e.g. up/down). The direct use of low-level descriptors from the inertial sensors or Kinect appeared not suitable to get a satisfying degree of generalization. Nevertheless, it appeared that participants were sharing similar 'frequency' behaviors for a given sound. For example, sounds with continuous pitched variations commonly induce smooth and slow gestures, repetitive sounds commonly induce synchronous oscillatory gestures, and sound textures can elicit 'shaky' ('vibratory') movements that seems to be related to metaphoric representation of stochastic sound characteristics.

These qualitative results led us to consider first a gesture analysis of the SkAT-VG dataset in deriving *frequency modes* of the movement.

Specifically, we propose an approach to movement analysis based on time-frequency representations using the Continuous Wavelet Transform (CWT). This method is compared to the standard approach based on the Windowed Fourier Transform (WFT). The Continuous Wavelet Transform representation is further used to extract mid-level descriptors by tracking continuous frequency component. These mid-level descriptors are used to two different recognition tasks: sound categories and gesture primitives.

3.1 Identification of Gesture Primitives

We propose to articulate the identification of gestural primitives in the SkAT-VG imitation database from two perspectives: the manual annotation of the dataset from two different experimenters on a reduced set of categories of gestures, and the computational identification of primitives through clustering.

From the preliminary experiments described in this section, and from the observation of the participants, we derived a reduced set of gestural primitives. In particular, we propose 6 gestural categories defined by either their frequency content, or by the oscillatory behavior of the gesture. We annotated the entire dataset according to this proposed taxonomy, as further described in Section 3.1.1.

The second perspective is computational and consists in identifying gesture primitives from the dataset using clustering. The methodology for the computational identification of gestural primitives consists in deriving gesture-level descriptors for the imitations of the SkAT-VG dataset. Each gesture is therefore described with a finite set of features related to the frequency content of the gesture. We then apply K-Means clustering to identify similar gestures in the dataset.

3.1.1 Manual Annotation of the Dataset

Each gesture unit is first segmented in three phases, as described by Kendon [Ken04]: Preparation, Stroke, Recovery. A gesture unit is understood as the movement excursion from a relaxed (still) position to another.

- *Preparation* refers to the phase that leads from the relaxed position to the stroke.
- *Stroke* is where the actual expression of the gesture is accomplished.
- *Recovery* covers the phase from the stroke to the final relaxed state.

The *Stroke* is the phase that defines the singularities of the gesture unit. It is what makes us identified it as a gesture.

Considering the stroke phase, we propose to define Six different *primitives*, characterized by their frequency components as follows:

- *Steady* gestures that practically do not change during time. It ranges from purely steady gesture to really slow evolution.
- *Smooth:* movements that are fluid and gradual.
- *Dynamic:* abrupt, energetic and rapid actions.
- *Impulse:* single and sudden excitation.
- *Periodic:* encloses all motions that have a periodicity in time.
- *Shaky:* is a specific class of periodic gestures that involves the hand shaking.

These primitives cover the complete frequency spectrum of gestures found in the dataset. While steady, smooth and dynamic describe unit variations in time, periodic and shaky stand for repetitions and impulse for single and short excitation.

The interface used for the annotation consists on a Max patch where all the needed information is displayed. The annotators have access to the video of the performance as well as the gestures and vocal expression signals. Markers can be added directly on the gesture and vocal displays. A right box lists all the markers defined by its start point and duration. An additional space is reserved for the label.

The three sound categories have been annotated. The *abstract* sounds has been annotated by two annotators. The *interaction* and *machine* sounds have been annotated only by one annotator, but we plan to complete this annotation task with a larger numbers of annotators.

For the *abstract* sounds, the inter-rater agreement have been evaluated by a Cohen's kappa coefficient of 0.67, which indicates a substantial conformity. The confusion matrix showed that most confusions occur between the dynamic and smooth categories. This is due to the fact that the threshold to consider a gesture as fluid or energetic strongly depend on the annotator evaluation.

The Figures 11, 12, 13 show the primitives distributions for the three different sound categories, respectively. The good agreement between the annotators #1 and #2 can be seen in Figure 11.

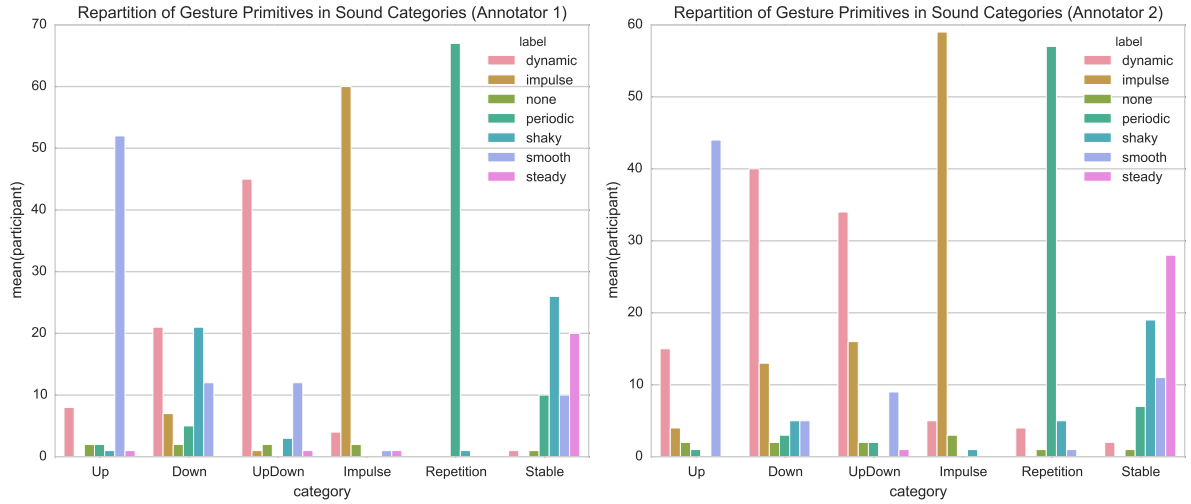


Figure 11: Primitives annotated for the abstract categories. Left: annotator #1. Right: annotator #2.

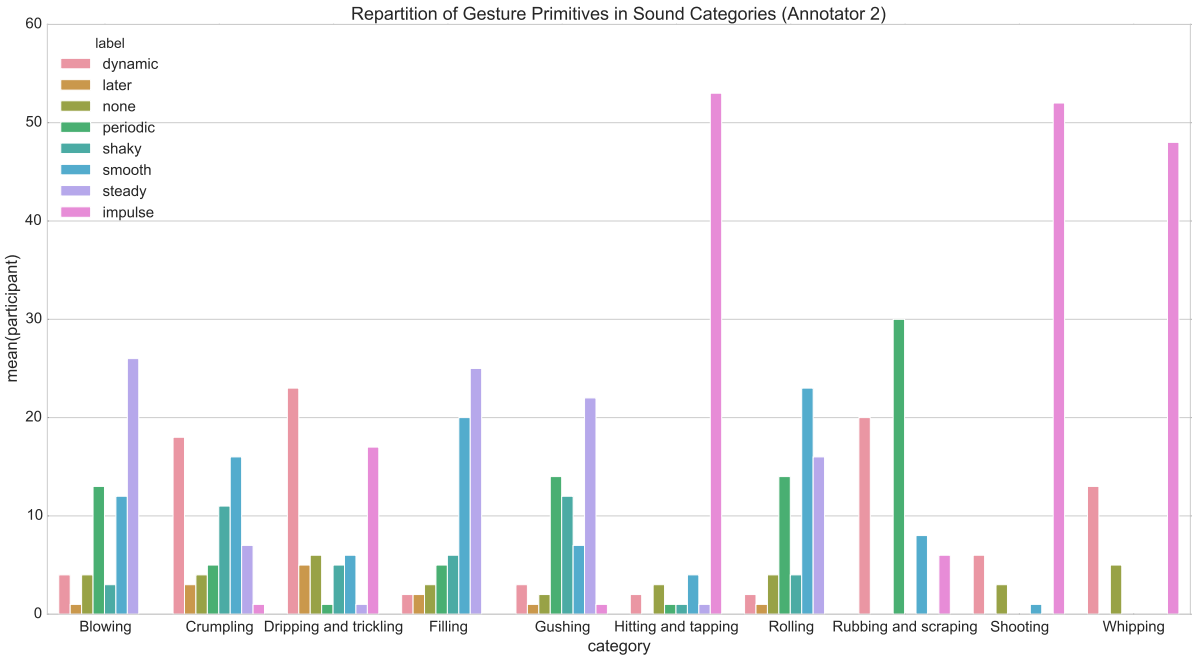


Figure 12: Primitives annotated for the interaction categories (annotator #2.)

3.2 Movement Analysis with the Continuous Wavelet Transform

We begin by introducing the Continuous Wavelet Transform (CWT) for time-frequency analysis of dynamic gestures. We propose an online implementation of the CWT with minimal delay and minimal complexity per frequency band.

Second, we propose a novel ridge tracking method for multiple fundamental frequency estimation using a multi-target tracking approach. We derive a particle filter implementation of a track-before-detect scheme for online estimation of ridge frequency, amplitude and variance.

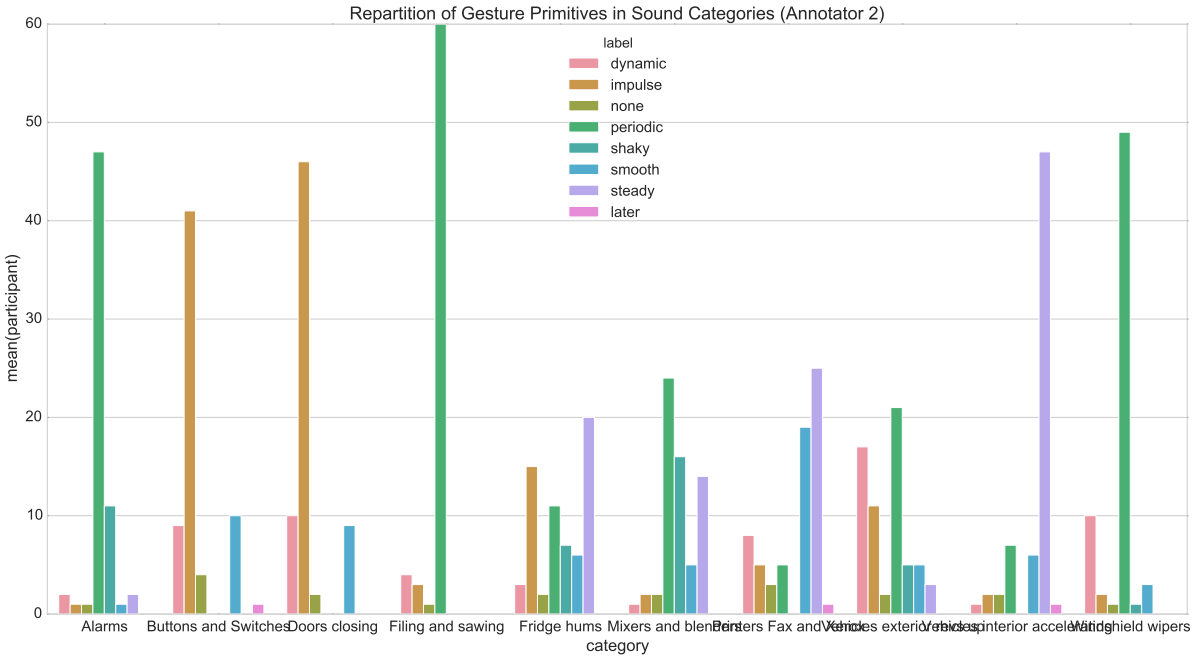


Figure 13: Primitives annotated for the machines categories (annotator #2.)

Third, we propose several gesture-level descriptors for the analysis of gesture primitives in the database of vocal imitation created at Ircam.

The overall analysis process is illustrated in Figure 14

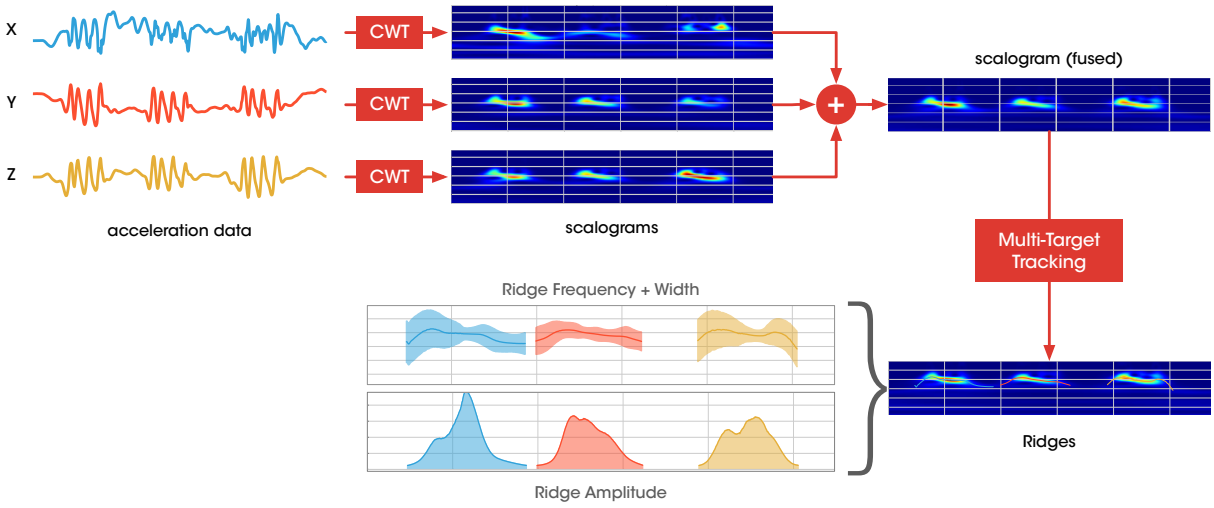


Figure 14: Overview of the analysis process.

3.2.1 Motivations

A rapid analysis of typical setups for capturing and representing movements further motivates the use of multi-resolution analysis. Typically, motion capture systems such as inertial sensors

or low-cost motion capture devices have frame-rates of about 100 to $500Hz$. Oscillatory movements are typically in the range of a few hertz, from $0.2 - 0.5Hz$ to $10 - 15Hz$ for smooth and periodic movements, and might be up to $50Hz$ for impacts. If one considers that the frequency resolution necessary to derive an accurate analysis around $1Hz$ should be at most of order $0.1Hz$, then the minimal window size required with the WFT is $10s$. In this case, any transient high-frequency phenomenon (of duration typically inferior to a second) will be blurred by the size of the analysis window. At the contrary, imposing a window size of $1s$ to guarantee an acceptable time localization restricts the frequency resolution to a bandwidth of $1Hz$, which is insufficient for capturing low-frequency phenomena. The multi-resolution analysis solution provided by the wavelet transform allows us to derive an arbitrary high localization both in time and frequency [Add05].

3.2.2 Introduction to the Continuous Wavelet Transform (CWT)

The Continuous Wavelet Transform (CWT) is a time-frequency analysis method allowing for optimal time-frequency localization [Add05, Mal08]. In this section, we consider how the wavelet transform can be used for characterizing dynamic behaviors in hand gestures. For more extensive tutorials on wavelet analysis, see for example [TC98, Add05, Mal08].

From Fourier to Wavelets In Fourier analysis, the spectrum of a signal is estimated by convoluting the signal by a set of harmonic plane waves. As all digital signals are finite, in practice the plane waves — or equivalently, the signals, — are multiplied by a window function to avoid artifacts due to border effects. The Windowed Fourier Transform (WFT) — or Short-term Fourier Transform (STFT) — allows to derive a time-frequency representation of signals by performing a Discrete Fourier Transform on a sliding window along the signal. The WFT therefore assumes a fixed window size, which determines the bandwidth of each frequency band in the time-frequency representation. One of the main limitation of the WFT is the inaccuracy resulting from the imposition of a scale or 'response interval' into the analysis [TC98]. Indeed, the WFT aliases all frequency components that do not fall within the frequency range of the window.

On the contrary, instead of assuming a fixed window size, the Wavelet Transform both translates and dilates a wavelet function with short-term influence. The dilation of the wavelet implies that the analysis windowed is expanded as the carrier frequency of the wavelet decreases. As a result of the Heisenberg's inequality, time-frequency resolution varies: the temporal windowing is short in high frequencies while the bandwidth is narrower in low frequencies.

3.2.3 Formulation

The Continuous Wavelet Transform (CWT) of a discrete sequence $\boldsymbol{x}_{1:N} = \{x_1 \cdots x_N\}$ sampled at period δt for a *wavelet function* Ψ_0 is defined as the convolution of the sequence with a scaled and dilated version of the base wavelet [TC98]:

$$W_n(s) = \sum_{n'=0}^{N} x_{n'} \Psi^* \left[\frac{(n'-n)\delta t}{s} \right] \quad , \quad \forall n = 1 \cdots N \tag{12}$$

where Ψ^* is the complex conjugate of the normalized wavelet:

$$\Psi\left(\frac{(n'-n)\delta t}{s}\right) = \left(\frac{\delta_t}{s}\right)^{1/2} \Psi_0\left(\frac{(n'-n)\delta t}{s}\right) \tag{13}$$

and s is the scale parameter. The time-frequency representation can be constructed by *translating* the wavelet along the time axis (varying n) and *dilating* the wavelet with the scale parameter s. By analogy with the term 'spectrogram' for the WFT, the 'scalogram' can be computed by taking the power representation of spectral information in the scale domain $|W_n(s)|^2$.

Offline Computation For offline estimation, the CWT can be efficiently evaluated using the Fast Fourier Transform to evaluate the convolution as a product in the spectral domain:

$$W_n(s) = \sum_{k=0}^{N-1} \hat{x}_k \hat{\Psi}^*(s\omega_k) e^{i\omega_k n\delta t} \quad \text{with angular frequencies} \quad \omega_k = \begin{cases} \frac{2\pi k}{N\delta t} & \text{if} \quad k \leq N/2 \\ - \frac{2\pi k}{N\delta t} & \text{if} \quad k > N/2 \end{cases} \tag{14}$$

For Most wavelets, there exist an analytical formulation of their Fourier Transform, which reduces the computational cost to a single inverse FFT per frequency band that simultaneously estimates the scalogram at all time steps.

Wavelet Functions While the WFT imposes windowed plane waves as set of base functions, wavelet analysis offers flexibility in the choice of the base functions for analysis. To be admissible as a wavelet, a function must meet three conditions: the must have zero-mean, finite energy, and their Fourier transform must be real and vanish in negative frequencies [Add05].

Several factors are to be considered in the choice of a wavelet basis [TC98]:

1. **Orthogonality**: Orthogonal wavelets produce the most compact signal representation, where the wavelet spectrum contains discrete blocks of wavelet power. However, orthogonal wavelets are sensitive to aperiodic shifts in the signal. Non-orthogonal bases are more suited to times series analysis: while it is redundant at large scales, the correlation of the wavelet spectrum is high at adjacent time, which tend to produce smoother variations of the scalogram. In this work, we focus on analyzing time series of movement descriptors and we only consider non-orthogonal wavelet bases.

2. Real vs **Complex**: A complex wavelet will return information about both amplitude and phase while a real wavelet is better suited to identify discontinuities. Considering that this research focuses on characterizing oscillatory behaviors in gestures, the choice of a complex wavelet is more relevant to our analysis.

3. **Width**: The resolution of a wavelet is characterized by the balance between its withs in real and Fourier Space. As we aim to capture both clear oscillatory behavior and transient phenomena, we need to select a wavelet offering a good compromise between time and frequency resolution.

4. **Shape**: Several shapes of the wavelet can be chosen according to the type of patterns to be found in the time series.

In this research, we mostly experiment with the Complex Morlet wavelet, which is further described in Section 3.2.3.

Choice of scales In comparison with the WFT, the CWT offer a great flexibility in the choice of the analysis domain. In particular, while the WFT restricts the analysis domain to the set of harmonic bands, one can choose a subset of arbitrary scales for wavelet analysis[4].

We follow a standard convention to distribute the scales as fractional powers of two:

$$s_j = s_0 2^{j/b} \quad , \quad j = J_{min}, J_{min}+1, \cdots, J_{max} \tag{15}$$

where b is the number of bands per octave and J_{min} and J_{max} can be specified from a target frequency range[5]. The smallest resolvable scale s_0 can be estimated from the equivalent Fourier frequency according to the Nyquist frequency $1/2\delta t$.

This representation is convenient to select the analysis domain, which specification is therefore reduced to a frequency range with a number of bands per octave.

Cone of Influence As for the WFT, the CWT is subject to edge effects resulting from the use of finite signals. While this problem can be partially resolved by padding with the edge values before performing the analysis, this process still introduces discontinuities at endpoints. The *cone of influence* is defined as the region where the edge effects become important, and can be defined in terms of *e-folding* time. "This *e-folding* time is chosen so that the wavelet power for a discontinuity at the edge drops by a factor e^{-2} and ensures that the edge effects are negligible beyond this point." [TC98].

In the remainder of this study, we consider a sufficient padding of the signals so that the scalogram can be estimated without edge effects at all scales. We also use the *e-folding* time as criterion for the online approximation of the CWT.

Morlet Wavelet The Morlet Wavelet — sometimes called Gabor Wavelet, — has the property of optimal localization in time and frequency. The complex Morlet Wavelet is defined as a plane wave modulated by a Gaussian window[6]:

$$\Psi_0(\eta) = \pi^{-1/4} \left(e^{i\omega_0 \eta} - e^{-\omega_0^2/2} \right) e^{-\eta^2/2} \tag{16}$$

where ω_0 is the carrier frequency of the Wavelet. The spectrum of the Morlet wavelet is a unit Gaussian centered around its carrier frequency. The temporal and spectral representation of the Morlet Wavelet are represented in Figure 15

[4]Note that for orthogonal wavelets, the choice of scales is restricted to a set of integer scales on a diadic grid. This limitation further motivates the use of non-orthogonal bases for movement analysis, that allow for using arbitrary scales and refine the resolution of the analysis

[5]For consistency, the upper bound for J_{max} is fixed to $b \log 2(N\delta t s_0)$

[6]We report here the 'complete' for of the Morlet Wavelet. As discussed by Addison et al. [Add05], considering that the correction term is negligible for $\omega_0 > 5.$, most articles in the literature use a truncated form for the Morlet wavelet: $\Psi_0(\eta) = \pi^{-1/4} e^{i\omega_0 \eta} e^{-\eta^2/2}$. However, in some cases it can be interesting to use smaller values of ω_0 in order to get a high temporal localization, as further discussed in Section 3.3

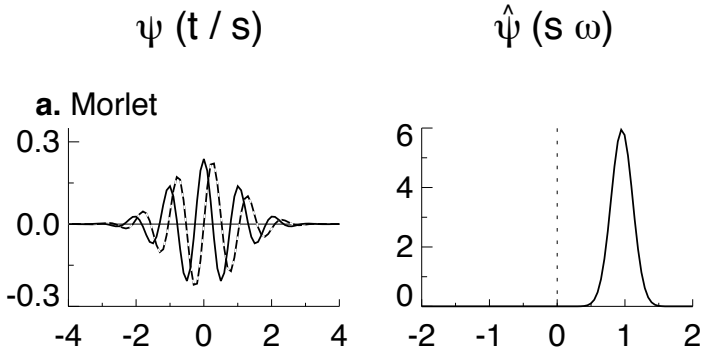


Figure 15: The Morlet Wavelet base. The plot on the left give the real part (solid) and imaginary part (dashed) for the wavelet in the time domain. The plots on the right give the corresponding wavelets in the frequency domain. For plotting purposes, the scale was chosen to be $s = 10\delta t$. from Torrence et al. [TC98]

The *e-folding* time for scale s with the Morlet wavelet is $\sqrt{2}s$, and the relationship between the scale and the equivalent Fourier frequency can be computed as

$$f = \frac{\omega_0 + \sqrt{2 + \omega_0^2}}{4\pi s} \tag{17}$$

Wavelet Analysis of Multidimensional Signals In the framework of the SkAT-VG project, we consider movements captured using either inertial sensors or markerless motion capture using Microsoft Kinect. Our approach to multidimensional analysis using the wavelet transform is based on a late fusion of the scalograms.

Consider the case of inertial sensors such as the MO [RBS+11] (Modular Musical Object) developed at Ircam. When analyzing dynamic movements with a sensor fixed on the wrist — as in the case of the imitation database, — our goal is to derive a representation of the frequency behavior that is invariant to the movement's direction and amplitude. In this case, we compute the scalogram for each axis of the accelerometer independently, and we represent gestures by the sum of the scalograms on each axis.

3.2.4 Examples

In this section, we report a set of illustrative examples that compare the proposed method with method based on the Windowed Fourier Transform. We start by presenting several results on synthetic data, and we illustrate the benefits on the method based on the Continuous Wavelet Transform with several examples from the imitation database.

The Figures 16, 17 clearly show that the oscillatory behaviors appear with a hight contrast in the Scalogramn (CWT) than in the Spectrogram (FFT). Moreover, the low frequency component is also better defined in the Scalogram.

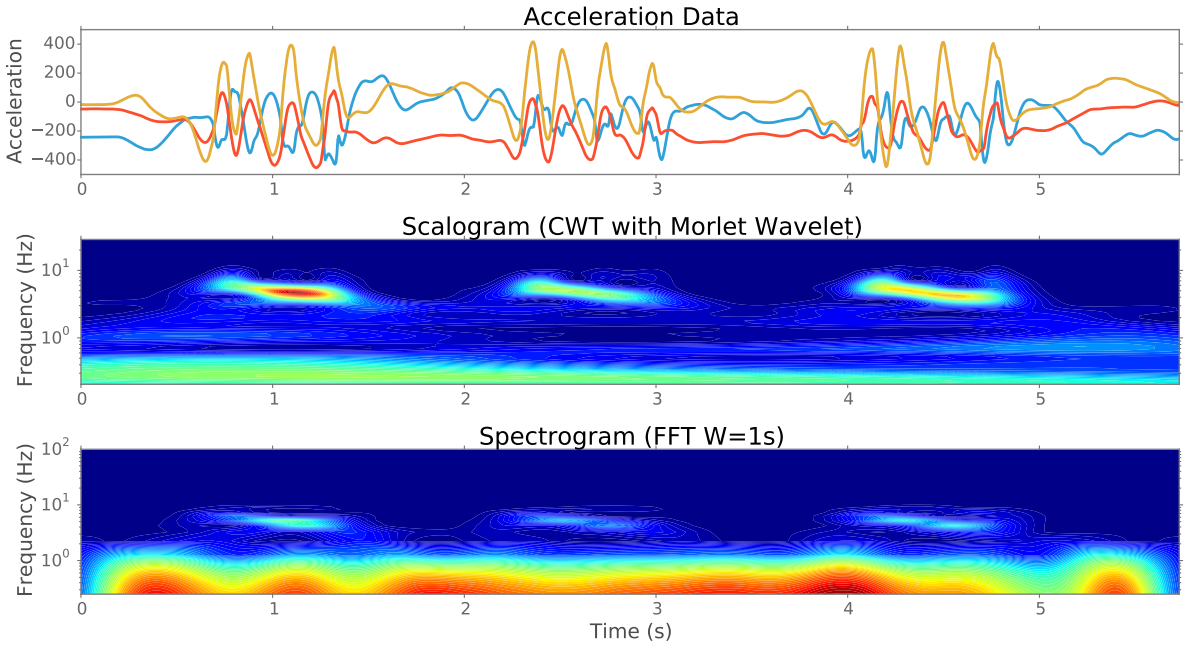


Figure 16: Contour plot of the time-frequency power spectrum of an acceleration signal using CWT and WFT. We used a window size of 1s for the WFT, and 8 bands per octave on frequency range $[0.2; 50]Hz$ for the CWT. For comparison, the images are remapped on Fourier frequencies.

3.3 Online Approximation of the CWT

We propose an online implementation of the CWT for interactive applications. Our approximation is based on a finite-length approximation of the wavelet depending on the cone of influence in each frequency band. We propose to use a computation window as small as possible at each scale to get a good approximation of the scalogram with a minimal delay in each frequency band. Additionally, we propose two optimization schemes based on a multirate representation of the wavelet and of the incoming signal.

3.3.1 Formulation

Our implementation is based on an approximation of the wavelet on a relevant cone of influence in each frequency band. In particular, we consider a finite-length windowing method where the window size is specified for each frequency band depending on the wavelet's energy decrease. The CWT is implemented as a filterbank with minimal delay per frequency band, and the computations are estimated in the temporal domain. At each new observation of the signal, we estimate only the central value of the scalogram on a sliding window with minimal size with regards to the e-folding time. This process is illustrated in Figure 19

For each scale s, the value of the scalogram with delay $N_s/2$ is estimated from a new observation value as

$$W_s[t - N_s/2] = \sum_{k=0}^{N_s} x[t - N_s + k] \cdot \Psi^* \left[\frac{(k - N_s/2)\delta t}{s} \right] \tag{18}$$

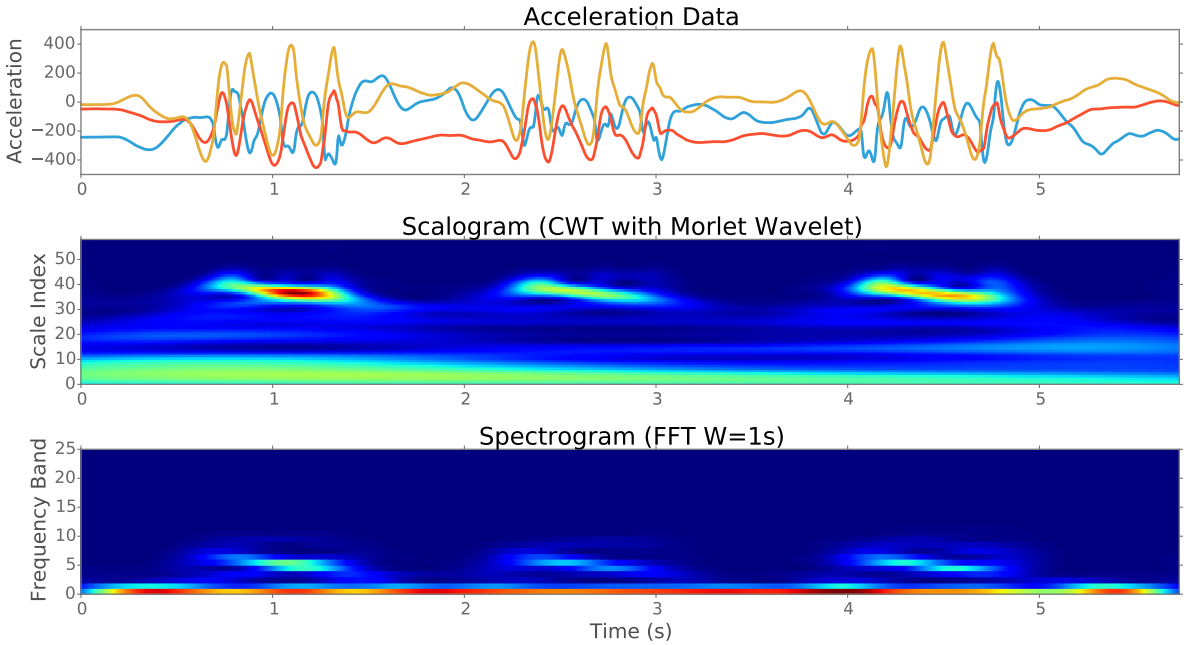


Figure 17: Time-frequency power spectrum of an acceleration signal using CWT and WFT. We used a window size of 1s for the WFT, and 8 bands per octave on frequency range $[0.2; 50]Hz$ for the CWT.

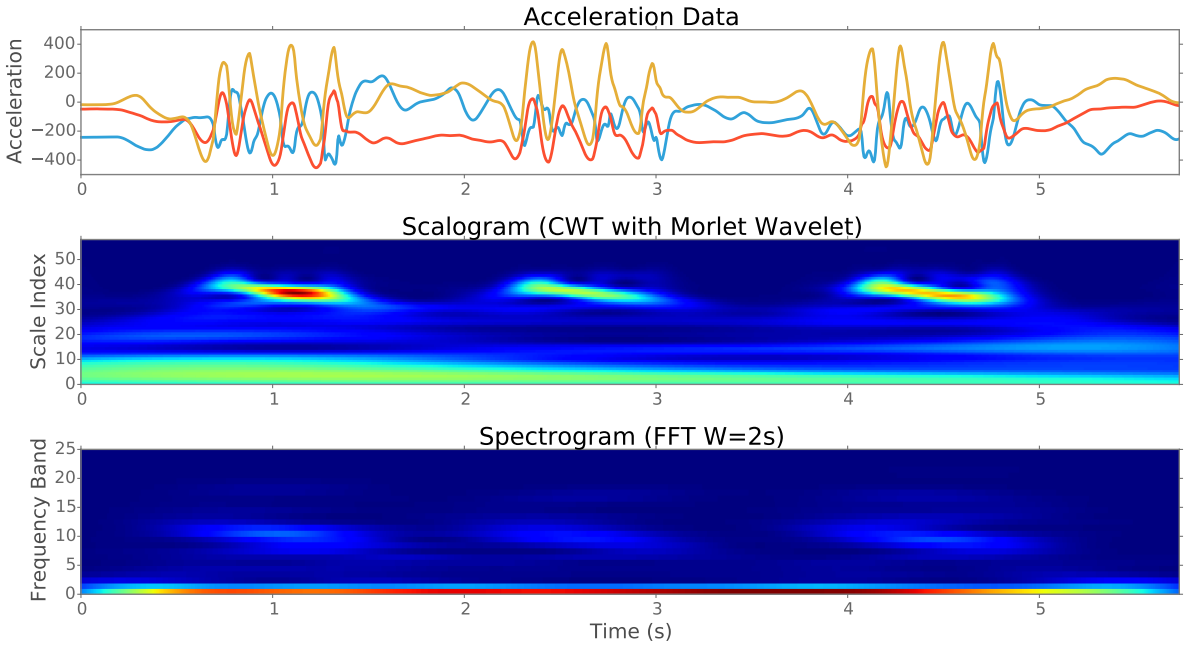


Figure 18: Time-frequency power spectrum of an acceleration signal using CWT and WFT. We used a window size of 1s for the WFT, and 8 bands per octave on frequency range $[0.2; 50]Hz$ for the CWT.

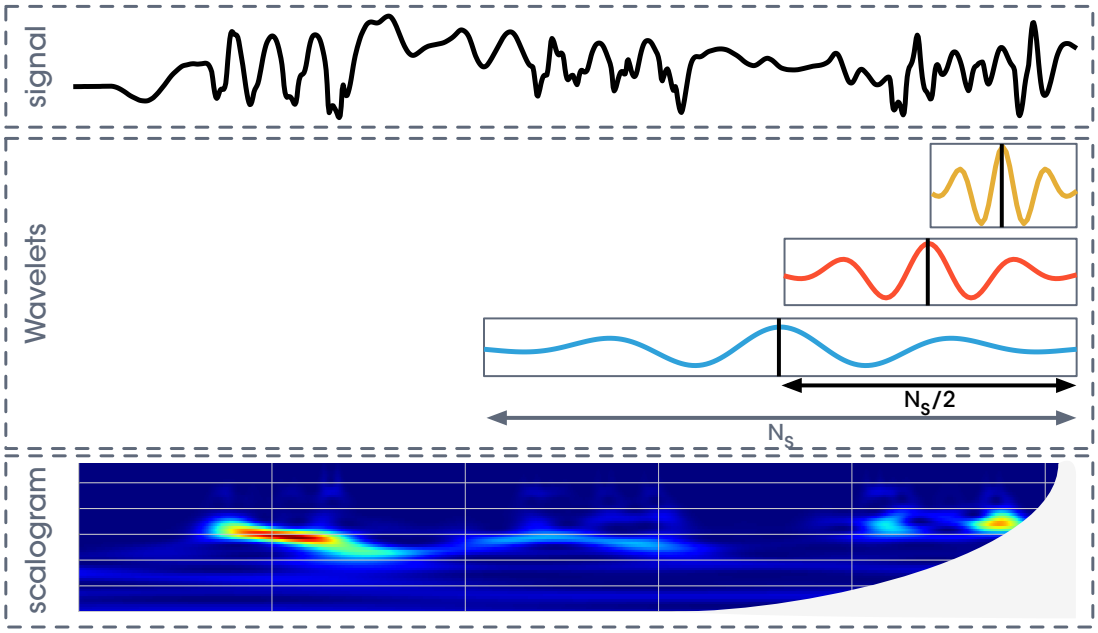


Figure 19: Illustration of the online implementation of the Continuous Wavelet Transform.

where the window size N_s can be estimated from the wavelet's e-folding time τ_s as $N_s = \lambda\tau_s/\delta t$ with λ a constant determined experimentally we call the *windowing factor*. For a Morlet wavelet with carrier frequency ω_0, the window size can be estimated at a given scale from the equivalent Fourier frequency f as

$$N_s = \lambda\sqrt{2} \cdot \frac{\omega_0 + \sqrt{2+\omega_0^2}}{4\pi f \delta t} \tag{19}$$

The complexity per frequency band is therefore N_s multiplications and $N_s - 1$ additions. For the Morlet wavelet with scales distributed in powers of two, the total complexity is exponential in the number of bands and can be estimated as

$$C = 2\sum_{j=J_{min}}^{J_{max}} \lambda\sqrt{2}s_0 2^{j/b} - 1 = 2\sqrt{2}\lambda s_0 \left(\frac{(2^{1/b})^{J_{max}} - (2^{1/b})^{J_{min}}}{2^{1/b} - 1} \right) \tag{20}$$

where b is the number of bands per octave and $s_0 = \frac{\omega_0+\sqrt{2+\omega_0^2}}{4\pi \cdot 2\delta t}$ is the highest resolvable scale.

On-line Implementation We implemented the online CWT as an external for Cycling'74 Max within the PiPo environment developed in the ISMM team at Ircam.Two modes are possible for the real-time analysis of movement data within Max.

In the first mode, the values of the wavelet power spectrum are outputted with a minimal delay in each frequency band. In this case, each band of the filterbank is evaluated with a different delay. While this approach does not guarantees a correct alignment of the different frequency bands, it provides a high reactivity in high frequencies, which can be interesting for

interactive applications where impulsive gestures should be identified with low-latency while low-frequency components are longer to establish.

In the second mode the wavelet spectrum is outputted with the same delay in all frequency bands, guaranteeing a correct alignment of the various components. However, this implies that the delay must be aligned on the largest delay corresponding to the lowest frequency component.

3.3.2 Optimization by Multi-rate Approximation

Computing the online CWT can be intensive at high framerates as the number of bands per octave increases, and as the analysis requires low-frequency components involving large window sizes. To alleviate this issue, we propose two optimization schemes of the online transform based on a multi-rate representation of the wavelets and of the signal.

Standard Optimization The number of computations per frequency band can be reduced by considering that low-frequency components can be approximated by a decimated version of the wavelet. We propose to decimate the wavelets by an integer factor depending on the ratio of their equivalent Fourier frequency with the Nyquist frequency. As a result, we guarantee that the wavelet's samplerate is sufficient to avoid aliasing the corresponding frequencies, while reducing the number of computation by an integer factor.

Note that in this case, the incoming signal is not decimated and we still evaluate the CWT at each frame for all frequency bands. To avoid aliasing in low frequencies, the signal is passed to a bank of low-pass filters for each decimation factor.

Aggressive Optimization Further optimization can be achieved by decimating not only the wavelet, but also the incoming signal. In this case, instead of evaluating the CWT at each frame for all frequency bands, we apply a similar decimation of the signal. The wavelet power spectrum is therefore evaluated at a lower framerate for low-frequency components, which provides a sparser representation of the scalogram.

Discussion As the standard optimization scheme keeps a frame-based computation of the scalogram, the estimation of the wavelet power spectrum remains smooth, which is particularly suited to interactive applications. On the contrary, aggressive optimization involves a severe downsampling of the incoming signal and the wavelet spectrum for low-frequency components is evaluated at larger time steps. This latter optimization scheme can be interesting when a sparse representation of the scalogram is desired.

In our experiments, *standard* and *aggressive* optimization techniques lead to a gain of a factor 20 and 80 in computation time, respectively. Note that the gain with standard optimization is made at the cost of an increased memory footprint, as it is required to store several versions of the incoming signal with different lowpass filtering. In both cases, the major drawback of the optimization methods is the introduction of an additional delay due to the low-pass filtering of the signal.

3.3.3 Examples

Figure 20 shows an example of the online approximation of the continuous wavelet transform computed over an acceleration signal from the SkAT-VG imitation dataset. The same plot with the realigned scalograms is showed in Figure 21, where the delay introduced by the online approximation is compensated in each frequency band, for comparison with the online estimate. While the online approximation without optimization introduces a delay, that increases as the scale increases, it can be seen that the distortion of the scalogram is relatively small compared with the offline estimate.

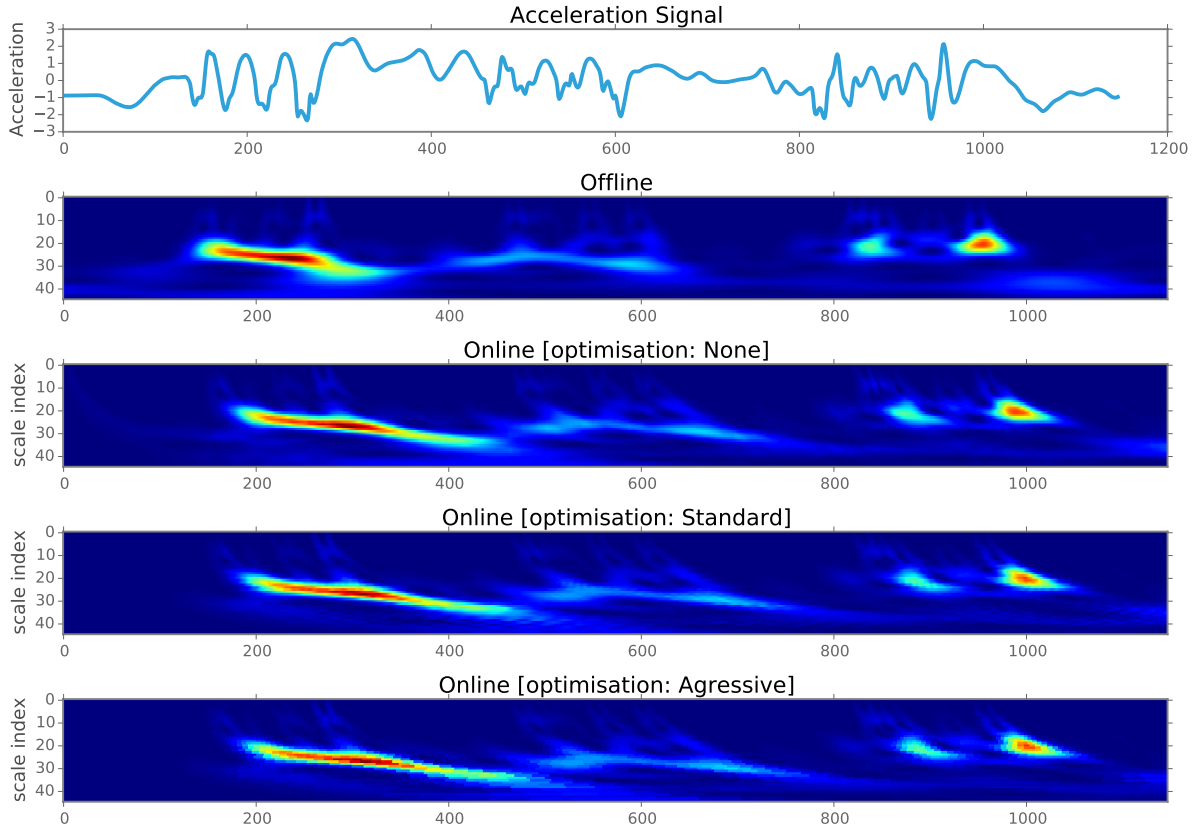


Figure 20: Example of online approximation of the continuous wavelet transform computed over an acceleration signal from the SkAT-VG imitation dataset. The approximation was computed with carrier frequency $\omega_0 = 5$ and *windowing factor* $\lambda = 3$.

3.3.4 Experimental Analysis

We evaluated the quality of the approximation of the scalogram using the online implementation of the CWT. Table 15 reports the Normalized Mean Squared Error (NMSE) or the online approximation of the scalogram as a function of the windowing factor. Each error is estimated with respect to the offline estimation of the scalogram, and is averaged over all imitations from all participants on the *Abstract* sound family. For comparison with the true estimate, the estimation delay of the power spectrum was compensated in each frequency band.

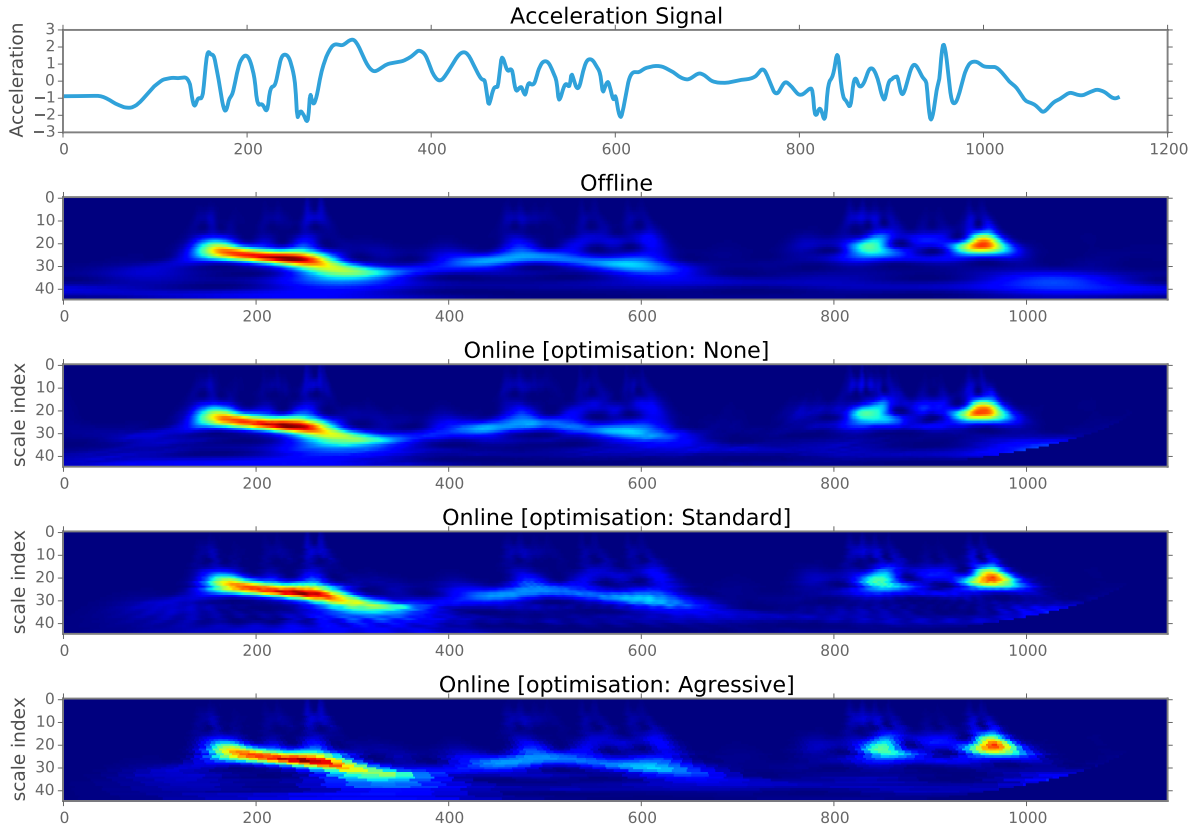


Figure 21: Example of online approximation of the continuous wavelet transform computed over an acceleration signal from the SkAT-VG imitation dataset. The delays in each frequency band are compensated for comparison with the offline estimate. The approximation was computed with carrier frequency $\omega_0 = 5$ and *windowing factor* $\lambda = 3$.

A relative window size of 3 — with respect to the e-folding of the wavelet in each frequency band, — provides a NMSE inferior to 5%. This approximation is sufficient in most interactive applications, as the approximation of the scalogram is not distorted.

3.4 Towards Multiple-mode Frequency Tracking

The wavelet spectrum representation of movement data provides rich information about the frequency content of dynamic gestures. In particular, periodic and impulsive gestures result in clear ridges in the scalogram centered around the fundamental frequency of the movement.

To further abstract the description of oscillatory gestures, we now investigate the quantitative measurement of gestures' frequency characteristics, in particular through frequency ridge tracking. The process is analogous to partial tracking for audio signals, where the spectral space is projected in the wavelet scale domain rather than the Fourier domain for FFT-based audio analysis. We reviewed in Considering the vast amount of work in multi-target tracking in computer vision and radar applications, reviewed below, we consider the implementation of a multi-target tracker for characterizing multiple frequency modes in dynamic gestures.

Table 15: Normalized Mean Squared Error of the online approximation of the scalogram compared with the standard estimate. Results are averaged over all imitations from all participants on the *Abstract* sound family. The results are displayed according to the size of the approximation window relative to the wavelet's e-folding time in each frequency band.

Windowing Factor	**NMSE (%)**	**Computation Time (10^{-3} ms/loop)**
1	54.676228	1.867
2	21.404965	3.687
3	4.632857	5.502
4	1.293430	7.362
6	0.008430	11.035

3.4.1 Multi-target Track-before-detect in the Wavelet Domain

Our approach follows recent developments in multi-target tracking in Radar tracking and computer vision systems. In particular, we consider track-before-detect (TBD) approaches to multiple object tracking where the target detection and tracking is performed jointly [QZLL15, SB01, BD04, EPS14]. Such approaches differ from standard detect-before-track methods where a set of potential targets are first detected — e.g. from image segmentation in the case of object tracking, — and then filtered using dynamic system modeling.

In TBD, the decision is made at the end of the processing chain, and therefore integrates all information over time. As a result, the tracking is less sensitives to errors in the detection step. Moreover, recursive approaches to TBD avoid the classical problem of data association between tracked targets and detected candidates: no thresholding is necessary, which removes the need for explicit association. Finally, it makes it possible to estimate additional parameters about the target, such as its intensity or size.

Problem Definition We aim at simultaneously tracking a maximum number K of frequency components of the movement — thereafter called 'targets', — from measurements of the wavelet power spectrum computed from one or several sensors. We assume that each target $k \in \{1, \cdots, K\}$, if present, is specified by a time-varying position $x^{(k)}$ in the wavelet scale domain — which is associated with the movement's fundamental frequency, — and a time-varying amplitude $a^{(k)}$. We assume that each frequency component is projected over the wavelet spectrum as a Gaussian, centered around $x^{(k)}$ and spread on the wavelet space with width $\Sigma^{(k)}$. Our goal is to estimate, from measurements of wavelet power spectrum, the number of components present at the current time step as well as their characteristic parameters $\left\{f^{(k)}, a^{(k)}, \Sigma^{(k)}\right\}_{k=1}^{K}$.

System Dynamics We assume that the position of each target evolves according to linear dynamics. We further assume a constant velocity model, meaning that, on a short time scale, each ridge evolves linearly in the wavelet domain.[7] For each target $k \in \{1, \cdots, K\}$, we assume

[7] We assume that the scales are distributed as fractional powers of 2, meaning that the associated distribution in the Fourier domain is logarithmic in frequency.

the following time-invariant state equation:

$$\boldsymbol{s}_{t+1} = \boldsymbol{f}(\boldsymbol{s}_t) + \boldsymbol{g}(\boldsymbol{s}_t)\boldsymbol{w}_t \tag{21}$$

where the process noise $\boldsymbol{w}_t$ is assumed to be standard Gaussian white noise. The hidden state $\boldsymbol{s}_t$ at time step t for a given target is composed by the target's position, velocity, amplitude, and spread:

$$\boldsymbol{s}_t = \begin{bmatrix} x_t \\ v_t \\ a_t \\ \Sigma_t \end{bmatrix} \tag{22}$$

The system dynamics function $\boldsymbol{f}$ and process noise input model $\boldsymbol{g}$ can be defined under a time-invariance assumption as

$$f(\boldsymbol{s}_t) = \begin{pmatrix} 1 & \delta t & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \boldsymbol{s}_t \quad , \quad g(\boldsymbol{s}_t) = \begin{pmatrix} \sigma_x \delta t^3/3 & \sigma_x \delta t^2/2 & 0 & 0 \\ \sigma_x \delta t^2/2 & \sigma_x \delta t & 0 & 0 \\ 0 & 0 & \sigma_a \delta t & 0 \\ 0 & 0 & 0 & \sigma_\Sigma \delta t \end{pmatrix} \tag{23}$$

where δt is the sampling period and σ_x, σ_a and σ_Σ are the expected maximum acceleration of the target, maximum change in amplitude, and maximum change in spread.

Measurement Model By analogy with radar sensors or optical system, we consider the measured power as reflected on a unidimensional observation space in the wavelet domain. A measurement $\boldsymbol{z}_t$ therefore consists of N power measurements z_t^i in the scale domain, where N is the number of bands of the CWT. In a multi-target setting, we hypothesize that the power measurements in the wavelet spectrum result from the superposition of the power emitted by each target, if present, yielding the following form:

$$\boldsymbol{z}_t = \sum_{k=1}^{K} \delta_k \boldsymbol{h}\left(\boldsymbol{s}_t^{(k)}\right) + \boldsymbol{n}_t \tag{24}$$

where δ_k is a binary indicator variable specifying if target k is present, and $\boldsymbol{h}\left(\boldsymbol{s}_t^{(k)}\right)$ is the reflected power of the kth target in the wavelet space, assumed to be Gaussian:

$$h^i\left(\boldsymbol{s}_t^{(k)}\right) = \frac{a_t^{(k)}}{\sqrt{2\pi\Sigma_t^{(k)}}} \exp\left[-\frac{\left(x_t^{(k)} - i\right)^2}{2\Sigma_t^{(k)}}\right] \quad , \quad \forall i \in \{0, \cdots, N-1\} \tag{25}$$

Track birth and Death We propose to model the possibility of birth and death of the frequency ridges using a jump Markov process, as often proposed in multi-target tracking systems [BD04, EPS14]. Instead of using a single *mode* state variable, as proposed by Boers et al. [BD04], we keep a vector of binary indicator variable $\boldsymbol{\delta} = [\delta_1, \cdots, \delta_K]^\top$. We assume that track birth and death are ruled a first-order Markov process where the probability of switching between different *modes* — i.e. configurations of the targets present at a given time

steps, — only depends on the mode at the previous time step. Constraints on track birth and death are then specified through a transition matrix that allows the birth of a new track with probability p_{birth} and the death of a new track with probability p_{death}. Additionally, it prevents the simultaneous birth and death of two tracks as well as the simultaneous birth of several tracks, in order to avoid target switching.

3.4.2 Particle Filter Implementation

We derived a a Bayesian solution to the estimation problem introduced in the previous section. While the state dynamics are linear and could be optimally solved analytically, the non-linear for of the observation model makes exact inference intractable. The track-before-detect can be solved using Sequential Monte Carlo estimation — also called particle filtering. Our particle filter implementation follows a standard Particle Filter implementation, as described in [BD04]. A example of the Ridge tracking from the scalogram is shown in Figure 22.

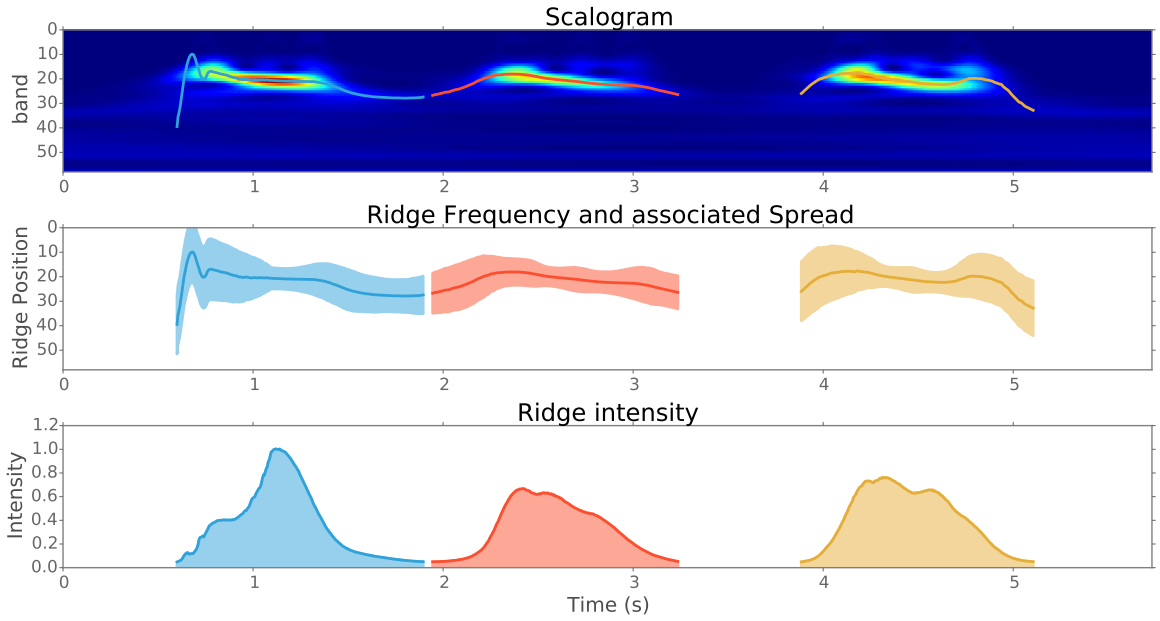


Figure 22: Example of Ridge tracking from the scalogram. The top plot represents the scalogram computed from the 3D acceleration of an example gesture from the SkAT-VG imitation dataset. The middle and bottom plots respectively represent the tracked ridge frequency and intensity. We used a single target with 100 particles.

3.5 Wavelet-based gesture descriptors

The analysis of the SkAT-VG imitation database requires deriving high-level representations of gestures. In this section, we describe two approaches to the representation of gestures based on gesture-level descriptors derived from wavelet analysis. The first approach is based on a description of the scalogram of an entire gesture by its normalized moments in the wavelet

spectrum domain and in the temporal domain. The second approach draws upon the multi-target tracking scheme introduced in Section 3.4. In this case we average the various ridges extracted from the tracking process to identify a fixed set of oscillatory modes in each gesture.

3.5.1 Spectral and Temporal Moments

We propose to describe the scalogram at a high level from the distribution of the wavelet power spectrum and of the energy envelope. The global wavelet power spectrum for an entire gesture can be obtained through the time-averaged scalogram (Figure 23, right):

$$\bar{P}(s) = \sum_{n=0}^{N-1} |W_n(s)|^2 \quad , \quad \forall s \in \{s_{min}, \cdots, s_{max}\} \tag{26}$$

Similarly, the energy envelope of the gesture can be obtained from the scale-averaged scalogram (Figure 23, bottom):

$$\bar{E}_n = \sum_{s=s_{min}}^{s_{max}} |W_n(s)|^2 \quad , \quad \forall n \in \{0, \cdots, N-1\} \tag{27}$$

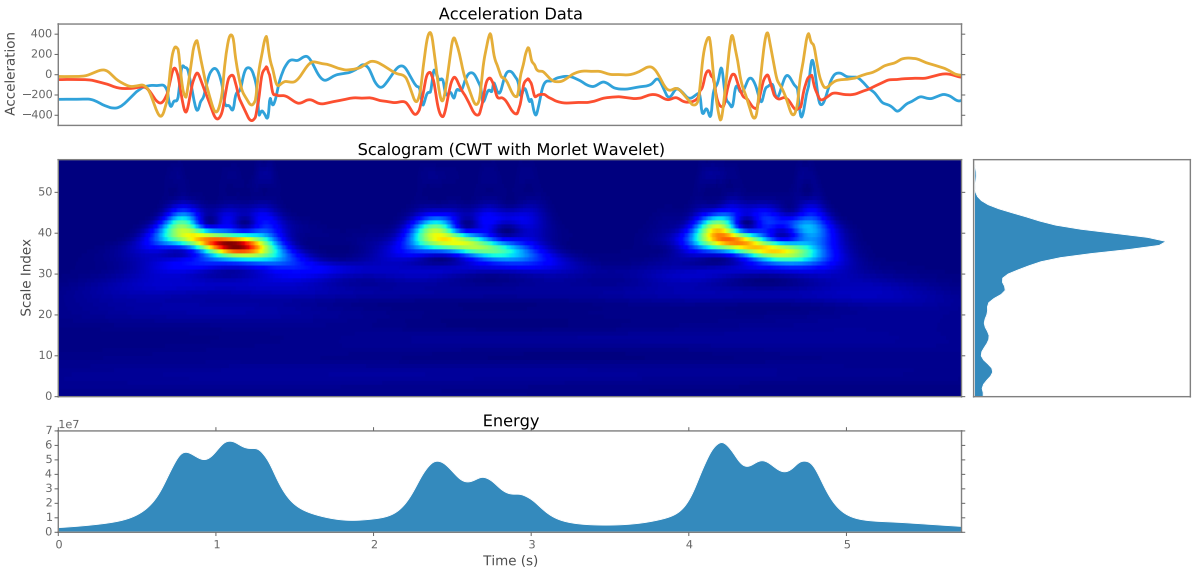


Figure 23: Description of a gesture through its scalogram's temporal and spectral moments.

We further abstract the temporal and spectral information by taking the normalized moments of each distribution, raising 8 parameters for the description of a single gesture: *Spectral Centroid, Spectral Variance, Spectral Skewness, Spectral Kurtosis, Temporal Centroid, Temporal Variance, Temporal Skewness, Temporal Kurtosis*. While this description remains simple, it is invariant to the scale of the movement data. This invariance is essential to the consistency of the analysis, considering that participants' energy in the gesture performance is extremely variable in the database.

3.5.2 Average Ridge Descriptors

A similar description can be derived from the more elaborate features obtained through the online tracking of frequency ridges. As result of the tracking process, several target ridges are identified and their *amplitude*, *frequency*, and *variance* are continuously estimated along the gesture. This allows to separate more clearly the contribution of concurrent frequency modes in the gesture.

Again, we can consider time-averaged ridge descriptors as the basis for a mid-level description of gestures. In particular, for a set of K ridges, we can compute $3K$ parameters per ridge as:

$$\begin{array}{ll} \bar{F}^{(k)} & = \sum_{n=0}^{N_1} F_n^{(k)} \quad \forall k \in \{1, \cdots, K\} \\ \bar{A}^{(k)} & = \sum_{n=0}^{N_1} A_n^{(k)} \quad \forall k \in \{1, \cdots, K\} \\ \bar{W}^{(k)} & = \sum_{n=0}^{N_1} W_n^{(k)} \quad \forall k \in \{1, \cdots, K\} \end{array} \tag{28}$$

The description of the temporal envelope can be obtained independently for each ridge from the estimated ridge amplitude $A_{1:N-1}^{(k)}$.

3.6 Clustering of Wavelet-based gesture descriptors

3.6.1 Method

We propose to identify gestural primitives through the clustering of Wavelet-based gesture-level descriptors. For this purpose, we run K-Means clustering on all recordings of the *Abstract* family of sounds. Although manual annotation resulted in 6 main categories, we used 8 clusters in the clustering algorithm to account for outliers in the dataset.

We propose two evaluations of the clustering process. The first evaluation is qualitative and consists in visualizing the clusters in a descriptor space that spreads the imitation in a 2D space according to gesture-level descriptors. This allows for visualizing and qualitatively assessing the type of gesture present in each cluster. The second evaluation is based on standard scores for evaluating the consistency of the clustering process. We use the V-measure to evaluate the agreement between the clusters identified using the clustering algorithm and the labeling of each annotation [RH07].

3.6.2 Visualization of Clusters and Multimodal Recordings

We developed a graphical interface for the purpose of visualizing, annotating and performing a qualitative analysis of participants' multimodal recordings. Our tool, showed in Figure 24, was developed within the Cycling'74 Max software with the multimodal container and visualization tool MuBu [SRS+09]. The graphical interface allows to browse the collection of recording from the global descriptors of each recording — namely, the frequency and temporal moments of each recording's scalogram, — to select and display multimodal data. Accelerometer signals, along with their respective scalogram and temporal power envelope are displayed, and the audio-visual recording of the participant can be visualized. This interface allows for exploring the complete dataset according to different dimensions of the analysis. Moreover, it enables us to perform a qualitative analysis of the behaviors aggregated in each cluster.

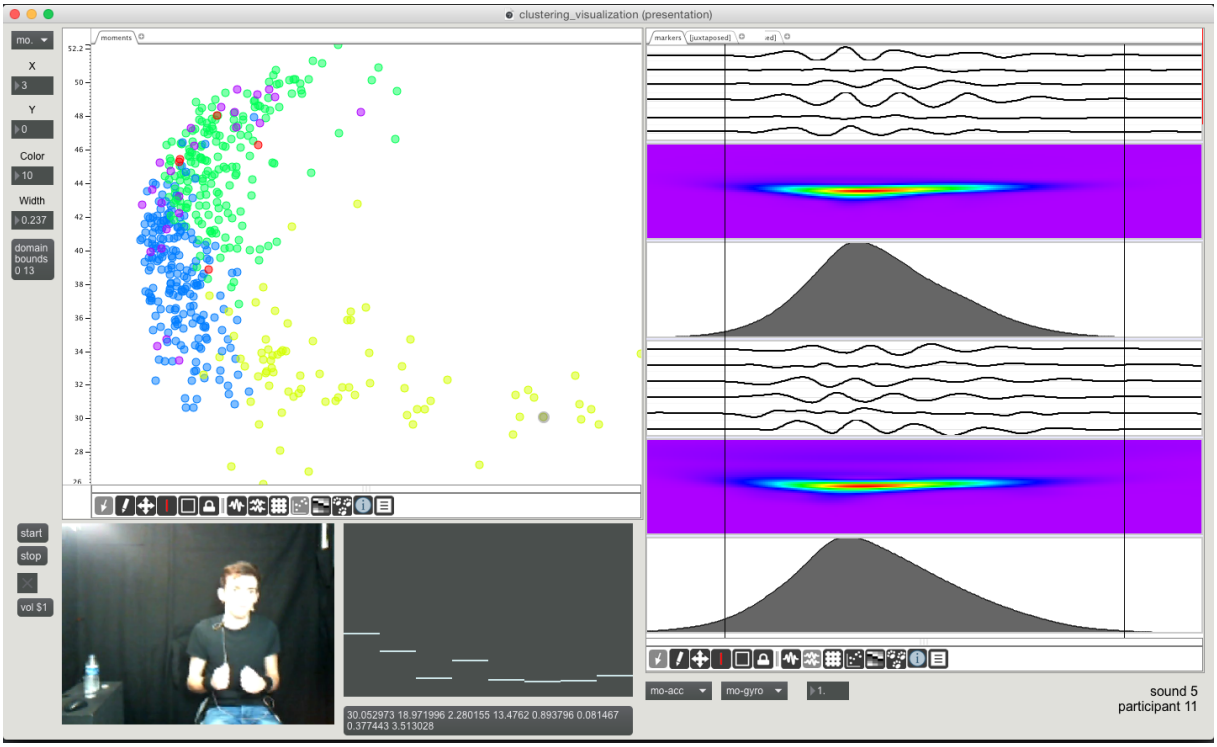


Figure 24: Screenshot of the interface for visualizing multimodal recordings of gestural and vocal imitations. The scatterplot (top left) displays all recordings according to a 2-dimensional descriptors space, with colors corresponding to clusters.

3.6.3 Quantitative Results

Table 16 details the V-measure obtained using K-Means clustering in comparison with each annotator's categorical annotation. The baseline denotes the V-measure computed between the annotators, which has a value of 0.5. This score emphasize a variability in the annotation between human annotators. The results clearly show that the gesture descriptors derived from the FFT gives lower match between the clusters identified computationally and the human annotation. For both the moments representation and the ridge tracking features, using only spectral content information is not sufficient to reach a significant agreement between the clustering and manual annotation. Temporal information, in particular the duration of each gesture, is essential to capturing the specificities of the gestural primitives. Indeed, while the frequency content of periodic and impulsive gestures lies in a similar range, they can be discriminated more easily from their temporal evolution.

An example of distribution of the recordings according to gesture-level descriptors is represented in Figure 25, along with the association of each recording to a cluster. The figure depicts the recordings in a 2D plane specified by the average frequency of the first 2 ridges, and the color of each recording represent its associated cluster or label. We observe an overlap between the categories specified by human observation and the computed clusters, in particular discriminating high-frequency gestures (bottom left), with varying duration (as indicated by superposed colors), and low-frequency gestures. The difficulty of performing clustering on the dataset arises from the continuity of the descriptor space: all recordings are continuously

Descriptor	Annotator 1		Annotator 2	
	CWT	WFT	CWT	WFT
Baseline: annotators	*0.50*	*0.50*	*0.50*	*0.50*
spectral moments	0.29	0.17	0.23	0.13
spectral & temporal moments	**0.38**	0.29	**0.32**	0.23
ridge freq	0.29	0.03	0.26	0.03
ridge freq + duration	**0.41**	0.14	**0.36**	0.14

Table 16: Evaluation of the clustering in comparison with the manual annotation. Each cell reports the V-measure between the clusters identified through K-Means clustering and the labels of each annotator, for a specific set of gesture-level descriptors. The baseline reports the V-measure between the annotations of each annotator for the *Abstract* family.

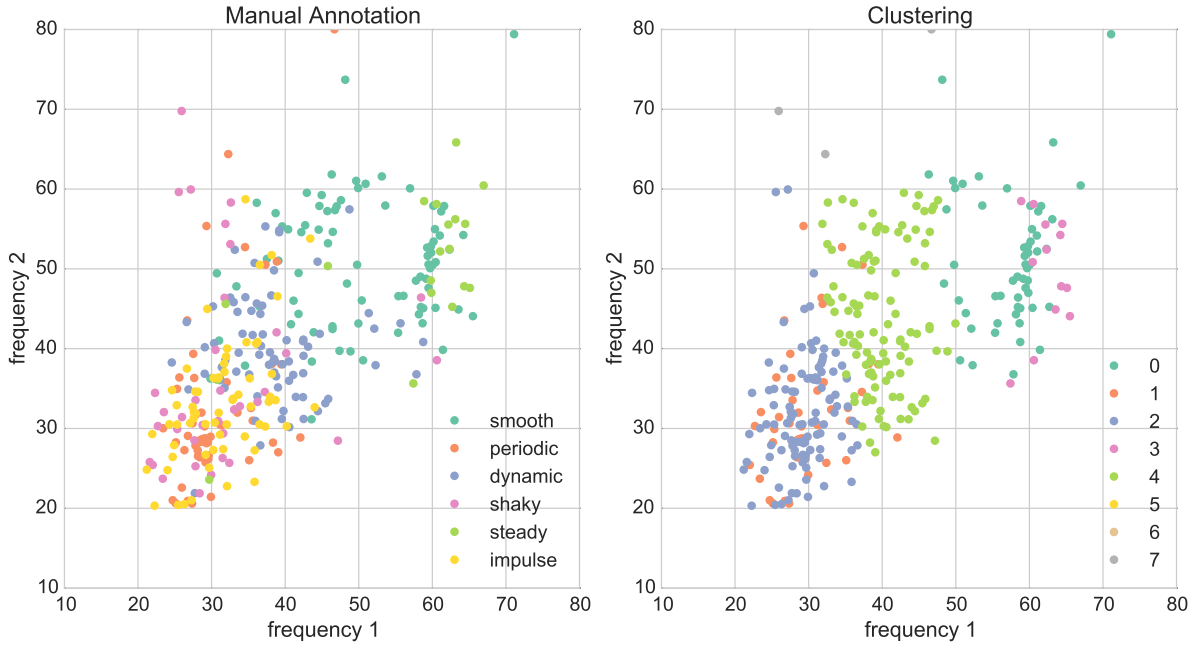


Figure 25: Example of identified clusters in a descriptor space. Each plot draws the recordings from the *Abstract* family according to the average frequency of each ridge in a 2-target tracking setting. Colors in the left plot represent the labeling of annotator 1 while the right plot represents the clusters identified with K-Means.

distributed in frequency and duration, which makes difficult the clear identification of the clusters boundaries.

3.7 Recognition of gesture primitives

In the previous section, we evaluate the recognition of the sound categories using gestural descriptors. In this section we evaluate, the recognition of the primitives defined by the annotators.

3.7.1 Method

The methodology for recognizing gestural primitives follows a standard 5-fold cross-validation where the 35 participants used in the experiment are randomly assigned to 5 groups. This guarantees that the gestures of a same participant cannot be found both in the training and test sets. The classification was performed using class-conditional Gaussian Mixture Models. We used 5 full-covariance Gaussian components per GMM, with variance regularization. We propose two approaches for the recognition of gesture primitives: an **offline** approach based on gesture-level descriptors, and an **online** approach using frame-level descriptors.

As baseline, we computed the accuracy score between the annotation of the two human annotators. Once again, the inter-annotator accuracy of 0.67 emphasize a relative agreement of the annotators, in particular between *smooth* and *dynamic* categories.

Offline Approach The offline approach is designed for batch recognition of the gesture primitives. It uses gesture-level descriptors and therefore requires well-segmented gesture recordings. We compare several gesture-level feature, derived either from the spectral and temporal moments of the scalogram, or from the ridge tracking algorithm. For the moments representation, we use the first 2 moments of the scalogram averaged either in time or scale, raisin 4 descriptors per gesture: the temporal and wavelet spectral centroid and variance. Gesture-level descriptors derived from the ridge tracking algorithm have a similar form, with simplified temporal modeling. The best scores were obtained by using only the duration information of each gesture with the average frequency of each ridge — amplitude and spread did not significantly improved the classification accuracy. For consistency, in multi-target tracking, the ridges are ordered in descending intensity.

Table 17 reports the classification accuracy with regards to each annotator's reference labels. With a 2s window size, the FFT gives significantly lower accuracy scores than the description based on the continuous wavelet transform. The results obtained using continuous ridge tracking are superior to the accuracy obtained using the moment-based description. This might indicate that ridge tracking is more robust to noise than the moments. However, no significant difference in accuracy is found between 1, 2 and 3 ridges for the recognition of gesture primitives on the Abstract categories, which might indicate that all gestures are sufficiently represented using a single frequency mode.

The confusion matrix associated with ridge-based classification is depicted in Figure 26. It indicates that most of the recognition errors occur between the *periodic* and *shaky* categories. This suggest that both categories could be merged: while the visual observation of some gestures tend to present random content, this randomness is associated with the hand and finger movements rather than with the wrist movements, that remain globally periodic. Merging the periodic and shaky categories lead to an improvement of the accuracy using all descriptors, that reaches about 80% using ridge tracking descriptors. The other sources of

	Annotator 1		Annotator 2	
Descriptor	CWT	WFT	CWT	WFT
moments	0.63	0.58	0.57	0.54
1 ridge	**0.68**	0.42	**0.62**	0.43
2 ridges	**0.67**	0.25	**0.59**	0.24
3 ridges	**0.66**		**0.59**	

Table 17: Classification accuracy of gesture primitives (steady, smooth, dynamic, impulse, periodic, shaky) using wavelet-based descriptors. The classification was performed using GMMs with 5 full-covariance Gaussian components. Each accuracy score was computed on 5-fold cross-validation on the imitations of *Abstract* sounds from 35 participants, where participants form the test set are absent from the training set.

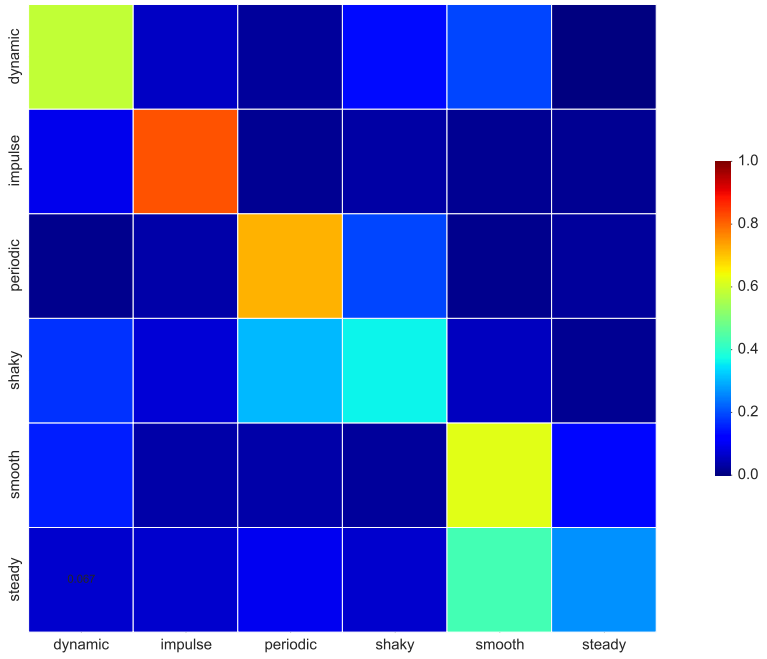


Figure 26: Confusion Matrix of the recognition of gesture primitives. The classification was performed using GMMs with 5 full-covariance Gaussian components from the gesture-level descriptors derived from ridge tracking with 2 target frequencies.

inaccuracy arise from the definition of the boundary between smooth and dynamic gestures. Once again, smooth and dynamic categories are defined using a threshold that is subjective to the annotator, while their union forms a continuous space with varying frequency and intensity.

Online Approach We designed an online approach to the recognition of gesture primitives where the recognition is performed from a continuous data stream. It aims at developing tools for continuous interaction where an accurate segmentation of the data stream might be difficult to estimate. In this case, the recognition is based on a representation of each gesture as a sequence of frames, each frame being described by higher-level features, either the spectral moments computed from each frame of the scalogram, or the frequency of each ridge computed from the online multi-target tracking.

	Annotator 1		
Descriptor	precision	recall	F1-score
moments	0.62	0.54	0.52
1 ridge	0.46	0.45	0.45
2 ridges	**0.65**	**0.61**	**0.61**
3 ridges	0.52	0.54	0.53

Table 18: Classification accuracy of gesture primitives (steady, smooth, dynamic, impulse, periodic, shaky) from frame-level descriptors. The classification was performed using GMMs with 5 full-covariance Gaussian components. Each accuracy score was computed on 5-fold cross-validation on the imitations of *Abstract* sounds from 35 participants, where participants form the test set are absent from the training set.

Table 18 details the classification accuracy for the different frame-level descriptions derived from moments and ridge tracking representations. The accuracy scores are lower than with gesture-level descriptors, which is consistent with the observation that gesture primitives are highly related to the duration of each gesture — which is now implicit in the recognition process rather than integrated as a specific descriptor. Interestingly, the best scores are now obtained using ridge tracking with 2 targets — i.e. when two frequency components are tracked in parallel. This might indicate that the discriminative power is improved when several parallel frequency modes are tracked in parallel. Once again, the classification accuracy of 65% is very close to the inter-annotator accuracy. Moreover, the accuracy can be improved to 75% when collapsing the *periodic* and *shaky* categories. These results are promising for interactive applications as the recognition process can be performed in real-time from a continuous data stream.

3.8 Recognition of sound categories for the Abstract family

3.8.1 Method

The methodology for recognizing sound categories primitives follows a standard 5-fold cross-validation where the 35 participants used in the experiment are randomly assigned to 5 groups.

This guarantees that the gestures of a same participant cannot be found both in the training and test sets. The classification was performed using class-conditional Gaussian Mixture Models, using frame-level descriptors We used 5 full-covariance Gaussian components per GMM, with variance regularization.

The same method will be used to evaluate the recognition of the gesture primitives, reported in Section 3.7

3.8.2 Results

	Annotator 1		
Descriptor	precision	recall	F1-score
moments	0.59	0.40	0.34
1 ridge	0.40	0.35	0.35
2 ridges	**0.53**	**0.50**	**0.49**
3 ridges	0.44	0.45	0.43

Table 19: Classification accuracy of sound categories for the *Abstract* family (up, down, up/down, stable, repetitive, impulse) using frame-level descriptors. The classification was performed using GMMs with 5 full-covariance Gaussian components. Each accuracy score was computed on 5-fold cross-validation on the imitations of *Abstract* sounds from 35 participants, where participants form the test set are absent from the training set.

Table 19 reports the precision, recall and F1-score on the task of sound category classification. The best score is obtained using ridge descriptors with two target frequencies, with a F1-score of 0.49.

Figure 27 reports the confusion matrix of the sound category classification task using ridge tracking descriptors with two target frequencies. The confusion matrix illustrates the similarities between gestural imitation strategies across sound categories. In particular, it highlights a high confusion between repetitive sounds and stable sound textures. This is consistent with the observation that noisy sound textures tend to induce shaky and periodic movements of the hands to emphasize the random aspect of the texture. Other sources of inaccuracy arise between Down and UpDown categories where the dynamics of the sound induce similar dynamics in gestural imitations.

3.9 Summary and discussions

The qualitative analysis of the IRCAM dataset clearly shown, as expected, a large variety in gesture characteristics among participants. This has been reported in the D4.4.1. While the

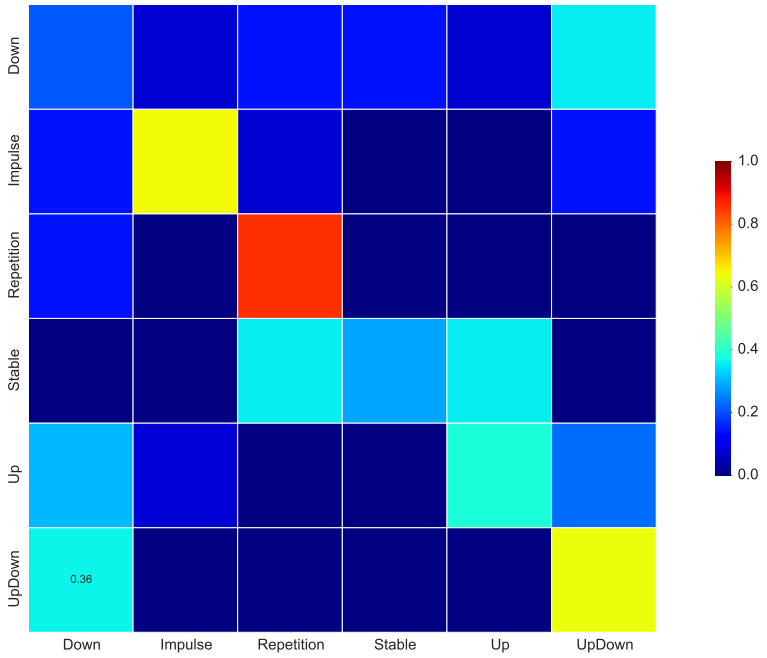


Figure 27: Confusion Matrix of the recognition of sound categories for the *Abstract* family. The classification was performed using GMMs with 5 full-covariance Gaussian components from the gesture-level descriptors derived from ridge tracking with 2 target frequencies.

direction and size of movements are not found consistent between users for a given category, similar 'gesture frequency modes' seemed to be found in several participants.

Therefore, we proposed to define several gesture primitives characterized by such different 'frequency modes', such as steady, smooth, dynamic, impulse, periodic and shaky. These primitives that were labeled manually. A good agreement between annotators were found for the abstract sound categories. This task should be further continued to complete the labeling with different annotators throughout the whole dataset.

In order to further evaluate quantitatively these ''frequency modes' and 'gestural primitives', we developed an analysis framework based on the Continuous Wavelet Transform (CWT). We first demonstrated that considering our application case, the Wavelet approach allows for quantifying frequency modes with a better compromise between time and frequency resolutions, compared to Windowed Frequency Transform.

Importantly, the analysis of the Wavelet scalogram allows to further extract gesture descriptors. We proposed to use statistical moments and ridges tracking using particle filtering. Finally, we also proposed a on-line implementation of the CWT analysis, which can be used in the SkAT-VG applications.

The different Wavelet gesture descriptors have been used in two different classification task. First, the recognition accuracy of the gesture associated to the Abstract Sound categories was found to reach 0.49 (F1-score). While this is clearly too low to be directly applied in applications, this recognition rate is much higher than expected. This confirms, as hypothesized, that the Wavelet based descriptors allows for capturing some frequency modes that are consistent among several participants. Second, the recognition accuracy of the manually

labelled gesture primitives were found to reach 0.67. The analysis of the confusion matrix revealed that are a refinement of gesture primitive definition should lead to higher results. These results confirms that the definition of such gesture primitives is promising to derive mid-level gesture descriptors.

Our proposed Wavelet framework to derive both offline and online gesture descriptors will thus be further evaluated and refined. The implementation of a multi-modal analysis with both vocalization and gestures represent one of the coming endeavors.

4 Audio recognition of vocal primitives

We have in the prediction of vocal primitives focussed on three articulatory classes, *phonation*, *myoelastic vibrations*, and *turbulence*. These classes can be used independently during vocalizations. Therefore, three different classifications were modelled and specific features were developed for each category. Here we had the opportunity to apply the recently developed Auditory Receptive Fields (ARF) toolbox for this work. Thus, both the articulatory modelling and the methods were developed for the first time within the project. The result with the best generalization was found using Partial Least Square Regression with a 3 or 5 components. The resulting classification accuracy was then for 10-fold cross validation 93.8 % for phonation, 85.8 % for myoelastic, and 79.7 % for turbulence.

4.1 Extracting data

WP5 uses the data set of articulatory annotations provided by WP3 as input for machine learning. To this end, the KTH team analysed the articulatory data and identified three contrastive articulatory parameters that seemed highly significant for imitation and that were amenable to analysis using audio analysis tools including the Auditory Receptive Fields (ARF).

- The presence vs. absence of *vocal fold phonation*.
- The presence vs. absence of slow *myoelastic vibration*.
- Third, the presence or absence of *turbulence* in the signal.

These three contrastive parameters make use of five of the eight articulatory annotation tiers in the database: Vocal fold phonation (Lar-VocFolds), Supraglottal laryngeal vibration (Lar-Sup), Nasality, Lip manner (Lip-Mann) and Tongue manner (Tongue-Mann). On the basis of various combinations of these tiers (and the values within each tier), two data sets were generated for each articulatory parameter; for example, a data set containing sound segments with slow myoelastic vibrations, and a data set containing segments in which slow myoelastic vibrations were absent.

4.1.1 Vocal fold phonation vs. No vocal fold phonation

The opposition *Vocal fold phonation* vs. *No vocal fold phonation* yielded two data sets. Segments in the *No vocal fold phonation* set can be defined as those segments in which a sound is produced but there is no vocal fold vibration. Thus, in addition to excluding segments with any manner of vocal fold vibration, segments involving a closure with the lips or the tongue are excluded even if they have no vocal fold phonation (because no sound is produced). To identify the relevant segments for the *No vocal fold phonation* set, four annotation layers have to be considered simultaneously: Lar-VocFold, Lip-Mann, Tongue-Mann and Nasality.

One could extract segments for the *Vocal fold phonation* using only the Lar-VocFolds annotation layer. However, using only one annotation layer to generate segments for the *Vocal fold phonation set*, while combining four annotation layers to generate segments for the *No vocal fold phonation* set would lead to a large imbalance in the number of tokens for the two sets (as well differences in the duration of the segments in each set). This is because the

combination of four annotation layers gives rise to a larger number of segments than does a single annotation layer. Therefore, the extraction for the *Vocal fold phonation* set made use of the same combination of layers as the *No vocal fold phonation* set.

The *Vocal fold phonation* set was populated with segments that had the following values in the Lar-VocFolds layer: "voiced" (which in turn collapses the original values modal and pressed), "falsetto" and "breathy". The Lar-VocFolds value "creaky" was not included in the set (5 occurrences in total). The "creaky" value describes creaky voice, which is not a good fit in the vocal fold phonation category in terms of frequency of vibration (it has more in common with the slower myoelastics in that respect).

At present, the *Vocal fold phonation* set contains 501 segments and the N*o vocal fold phonation* set contains 690 segments.

4.1.2 Slow myoelastic vs. Non-myoelastic

Two data sets were generated for the opposition *Slow myoelastic* vs. *No slow myoelastic*. Segments in the *Slow myoelastic* set were extracted from a combination of four annotation layers: Lip-Mann, Nasality, Tongue-Mann and Lar-Supra.

The *Slow myoelastic* set was populated with segments that had the following values: "myoelastic_lax" (Lip-Mann), "velic_myoelastic" (Nasality), "myoelastic" (Tongue-Mann), and "aryepiglottal"/"ventricular" (Lar-Supra). The Lip-Mann value "myoelastic_tense" (6 occurrences) was not included in the *Slow myoelastic* set, as it describes a fast, bilabial vibration (very much like embouchure), which is not a good fit in the otherwise slowly vibrating myoelastic category.

The *No slow myoelastic* set includes segments in which a sound is produced and the sound is not a myoelastic sound. Thus voiceless occlusives are excluded from the set (no sound) as well as all myoelastics. The *No slow myoelastic* set was extracted from the same combination of layers as the *Slow myoelastic* set. Note, however, that voiced occlusives are included in the set but that these were extracted using another combination of layers: Lar-VocFolds, Nasality, Lips-Mann and Tongue-Mann.

At present, the *Slow myoelastic* set contains 232 segments and the *No slow myoelastic* set contains 908 segments.

4.1.3 Turbulent vs. Non-turbulent

The data sets for the opposition *Turbulent* vs. *Non-turbulent* were both extracted from the combination of four annotation layers: Lar-VocFolds, Nasality, Lip-Mann and Tongue-Mann.

The *Non-turbulent* set contains segments in which a sound is produced but the sound is not turbulent. Thus voiceless occlusives are excluded from the *Non-turbulent* set (because no sound is produced) as well as all segments with a "turbulent" value in the Lip-Mann or Tongue-Mann tiers. Also excluded are segments with an "abducted" value for Lar-VocFolds when both Lip-Mann and Tongue-Mann tiers have an "open" value (i.e. glottal friction as in h-like sounds).

Segments for the *Turbulent set* were extracted using the following values: "L-turbulent" (a value that collapses the Lip-Mann values "turbulent_spread", "turbulent_rounded" and "turbulent_labiodental"), and "turbulent" (Tongue-Mann). In addition, to capture glottal friction

(h-like sounds), the following combination of values from different tiers was used: "abducted" (Lar-VocFolds) + "L-open" (Lip-Mann) + "open" (Tongue-Mann). (Note that the "L-open" value collapses the Lip-Mann values "open_rounded", "open_half_rounded", "open_neutral", "open_spread" and "L-transition".)

At present the *Turbulent set* contains 606 segments and the *Non-turbulent* set contains 643 segments.

4.1.4 Final extraction

The above described procedure generated a list of segment data pointers and the final audio was extracted using a script in Matlab. This whole process was repeated twice since it turned out to be difficult to, in particular, define precisely the content of the negative group, i.e. the ones not containing the category in question in the first attempt.

The analysis of in particular slow myoelastic vibrations demands a certain time window. The duration limit of the included segments was set to 150 ms. This corresponds to three cycles of a vibration of 20 Hz. This made also the balance between the number of segments in the positive and negative category in each class more even. The final distribution of the three data sets are provided in Table 20. There is a reasonable even distribution of the number of segments across classes and subjects. However, note that the myoelastic and turbulence class have a larger portion of negative segments.

Classes:	**Phonation**	**Myoelastic**	**Turbulence**
Total n	438	434	453
Positive	204	147	190
Negative	234	287	263
Subject 1	116 (56/60)	118 (48/70)	122 (53/69)
Subject 2	100 (47/53)	91 (24/67)	101 (47/54)
Subject 3	124 (52/72)	127 (52/75)	128 (38/90)
Subject 4	98 (49/49)	98 (23/75)	102 (52/50)

Table 20: The distribution of the number of segments in each of the three articulation classes. Numbers in parenthesis refer to the number of positive and negative segments for each subject.

4.2 Auditory receptive fields toolbox

The audio examples were analysed using the auditory receptive fields toolbox (ARF toolbox; [LF15a], [LF15b]) implemented in Matlab. It is a new framework for analysing sounds mimicking the neuron functions in the auditory pathway. The first stage is to transform the audio into a time-frequency representation in form of a spectrogram. Its properties include logarithmic frequency bins, constant bandwidth and time-causal processing. In subsequent stages receptive fields defining a local area in time-frequency can be applied to the spectrogram. Depending on the shape and size of the receptive fields, different properties can be enhanced such as the onsets, partials, and formants (see [LF15a]). The ARF toolbox was recently developed and this is the first time it has been applied to a practical problem.

4.3 Features

4.3.1 Phonation features

1. Transformation from audio to spectrogram using receptive fields models.

The spectrogram was calculated using the ARF toolbox. We used a time-casual kernel defined by seven recursive filters in cascade with a logarithmic distribution of the temporal scale level with $c = \sqrt{2}$, see Fig. 28 (and Fig. 5 in [LF15a]).

The width of the kernel in terms of its standard deviation was in the middle frequency range eight cycles of the centre frequency of each bin. Thus, in this range, the frequency bandwidth was constant with respect to logarithmic frequency. In the upper and lower part of the frequency range the standard deviation of the kernel was gradually flattened out to a constant value. In the lower range this prevented the kernel to become unrealistically wide for low frequencies.

The frequency bins were logarithmically spaced from midi note number 36 to 132 (approx. 65 Hz to 16.7 kHz) with a resolution of 48 bins per octave. The position in time of each bin was time-compensated using the inflection point of the kernel as the reference.

The magnitude of the spectrogram was converted to sound level with a range of 60 dB.

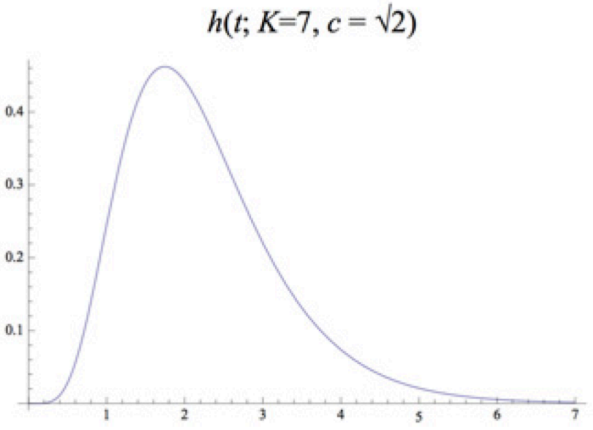


Figure 28: The time-casual kernel used for the spectrogram transformation.

2. Removal of silence before and after sound

The spectrogram was smoothed in frequency using receptive fields applied on the spectrogram. A discrete Gaussian kernel was used with the standard deviation in frequency of 3 semitones, corresponding roughly to critical bands, and with a standard deviation in time of 0.01 s.

A detection function was calculated by taking the maximum sound level across each frame. A fixed threshold was defined at -25 dB below the maximum sound level for the whole example. Anything that was below this threshold was removed for the initial and final part. The resulting duration in seconds was used as a feature in the classification.

3. Whitening in spectral dimension using an ARF Gaussian filter.

A smoothing filter was applied using a discrete Gaussian kernel with the standard deviation in frequency of 8 semitones. For each frame the filtered spectrum was subtracted from the

original spectrum using an offset of 3 dB and a total range of 50 dB from the maximum.

4. Enhancement of harmonic fundamental frequency

It was assumed that the spectrum of the phonation part of the sound was perfectly harmonic and the remaining part of the audio consisted of some kind of noise. The enhancement of the harmonic fundamental was done by adding translated spectrogram copies. These translations were done according to the harmonic series. For example, the spectrum translated one octave down was added to the original spectrum. In this way the fundamental was enhanced with the first partial. One example of the final harmonic enhancement is shown in Fig. 29.

5. Frame feature extraction

The following frame-based features were calculated from each frame of the enhanced fundamental: sound level and frequency of highest peak, sound level and frequency of second highest peak, difference between the two peaks, and mean sound level all peaks except highest.

6. Final features across frames

The statistics used for the final features were the upper quartile, standard deviation, and the mean of the difference between frames (mean derivative in time) that were calculated for each frame feature. This resulted in a total of 19 (6*3+1) features for the phonation class.

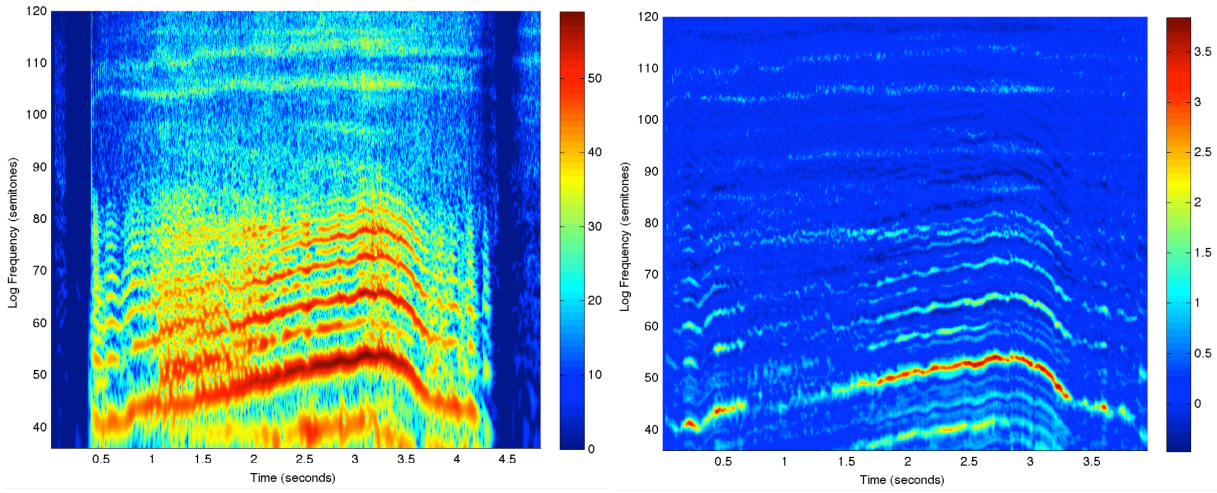


Figure 29: The original spectrogram (left) and the resulting enhancement of harmonic fundamental (right). The sound is an imitation of an accelerating lorry containing both phonation and turbulence.

4.3.2 Myoelastic features

The myoelastic features are calculated using two alternative approaches. The first method are detecting possibly irregular but cyclic variations in the upper part of the spectrogram adapting a previously developed method for vibrato detection. The other one is using the time representation of audio combined with auto-correlation. Both sets of features are used in the final prediction.

Myoelastic features using spectrogram

The same spectrogram was used as input that was calculated for the phonation features above. The following steps were applied:

1. Extract sound level curve for upper spectrum.

It was evident that the myoelastic vibrations were exhibited as sound level variations in the upper part of the spectrogram. Therefore, the sound level curve of all frequencies above 1 kHz was calculated using the median for each frame.

2. Detect regular amplitude variations

From the sound level curve cyclic variations were detected using the three-point method previously used for vibrato detection suggested by Prame [Pra94] and implemented by Erwin Schoonderwaldt [FSJ05]. Two separate analyses were done using different frequency ranges addressing both the relatively slow myoelastic vibrations by the tongue and lips around 30 Hz and the somewhat faster vibrations produced by various inner parts of the throat and tongue that are around 70 Hz but with a rather large span. For the slow variations a low pass filter with a cutoff of 50 Hz was first applied and the then detection was made with a range of 15 to 40 Hz. For the fast variations the cutoff was 220 Hz and the detection range was 40 to 200 Hz. This resulted in two sets of discrete detection points marking both each detected peaks and troughs for the two analyses.

3. Final features

The were four features calculated from the detected points for each analysis. The first two features consisted of the median across all detected points for the variation rate and amplitude. The third feature was a calculation of the total length of detected variations relative to the total length of the example. The fourth feature was the multiplication of the amplitude (No. 2) and the relative duration feature (No. 3). This resulted in a total of 8 myoelastic features.

Myoelastic features using time representation

1. Auto-correlation

Taking into account the notion that periodic sounds can be detected by correlating an audio file with itself in the time domain, auto-correlation was performed on the myoelastic +/- fragments. This was accomplished by creating an implementation of the YIN pitch detection algorithm [Rao11] [DCK02]. The necessary input values of minimum and maximum frequency, hop and block size as well as base frequency have been specified manually, and were tested with different values to ensure the best balance between speed and accuracy. Specifically, the minimum frequency was chosen to be 20 Hz, which is believed to be the lowest frequency of manifestation of any classification type. Conversely, the maximum frequency was set to 200 Hz, which no classification type is believed to exceed. Furthermore, hop and block sizes were set to 10 ms and 20 ms, respectively. While the former parameter speeds up the calculation when bigger, the latter speeds it up when smaller. Finally, the base frequency has been set to 30 Hz, which is believed to be the frequency most myoelastic samples occur at.

The auto-correlation itself compares every sample at index i with another sample at index i+tau, with tau being an offset vector ranging from tauMin to tauMax according to the maximum and minimum frequencies specified above. This procedure is executed separately for each frame. As a result, the auto-correlation matrix shows the cumulated differences within each

frame (columns) at every value in the vector tau (rows). The smaller the difference, the more the values correlate with each other. Fig. 30 shows an excerpt of the autocorrelation function of an arbitrary fragments first frame, with the resulting median value (smallest difference) of this particular frame as the red vertical line.

2. Frequency peak matrix computation

After having computed the auto-correlation matrix for a specific fragment, the highest and second highest peaks of the specified frequencies are identified based on the value of tau with the smallest difference in each frame. Much like the with auto-correlation matrix itself, this procedure is executed separately for each frame. The number of frames thereby depends on the input parameters mentioned above.

3. Frame feature extraction

A number of features are derived from the frequency peak matrices. For both the highest and second highest peaks, the following values are computed:

- median
- mean of the absolute value of variation (mean derivative in time)
- standard deviation

Additionally, the following global values are extracted:

- absolute value of difference between aforementioned median values
- amplitude under a Gaussian curve (with the base frequency as peak)

These eight distinct features as well as their identifiers are subsequently passed on to the calling method, in which they can be analysed. Fig. 31 shows the Gaussian curve of an arbitrary fragment, with the resulting median value as the red vertical line. The actual feature is the amplitude under the curve (in this example 0.7827).

4.3.3 Turbulence features

For the turbulence category we applied the following steps to the resulting spectrogram calculated in step two in the section above about phonation features:

1. Removal of spectral peaks using a smoothing filter

The smoothing filter was specifically designed to remove the harmonic partials considering the bandwidth of the spectrogram and the variation of the harmonic density across the spectrum. A filter with a varying window as a function of frequency was applied using the 15% percentile. It can be viewed as a development of median filtering used for separating harmonic and percussive content [Fit10] [EF15]. The percentile and window size across spectrum was chosen manually using the spectrogram displays in order to minimize the harmonic content while retaining most of the turbulence (noise), see Fig. 32.

2. Frame feature extraction (spectral bands)

The remaining spectrum is both divided in seven octave bands and in two bands (over or under 1kHz). For each band (and frame) the median value is calculated.

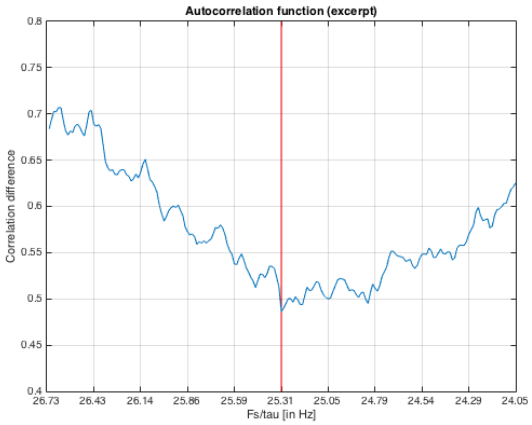


Figure 30: Excerpt of the autocorrelation function of an arbitrary fragment's first frame, with the resulting median value of this particular frame as the red vertical line.

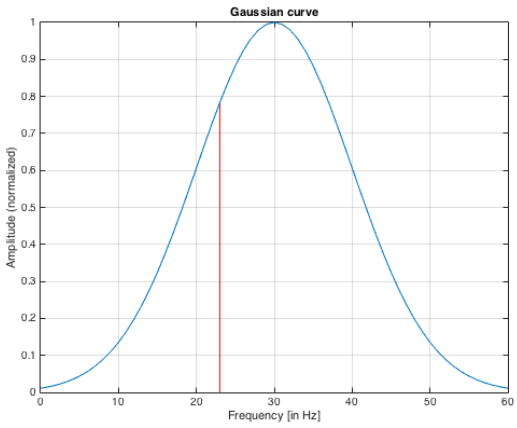


Figure 31: Gaussian curve of an arbitrary fragment, with the resulting median value as the red vertical line.

3. Final features across frames

The final features are currently calculated in the same way as for the phonation features. Thus using the upper quartile, standard deviation, and the mean of the difference between frames (mean derivative in time). This resulted in a total number of features of 27 (9*3) for the turbulence class.

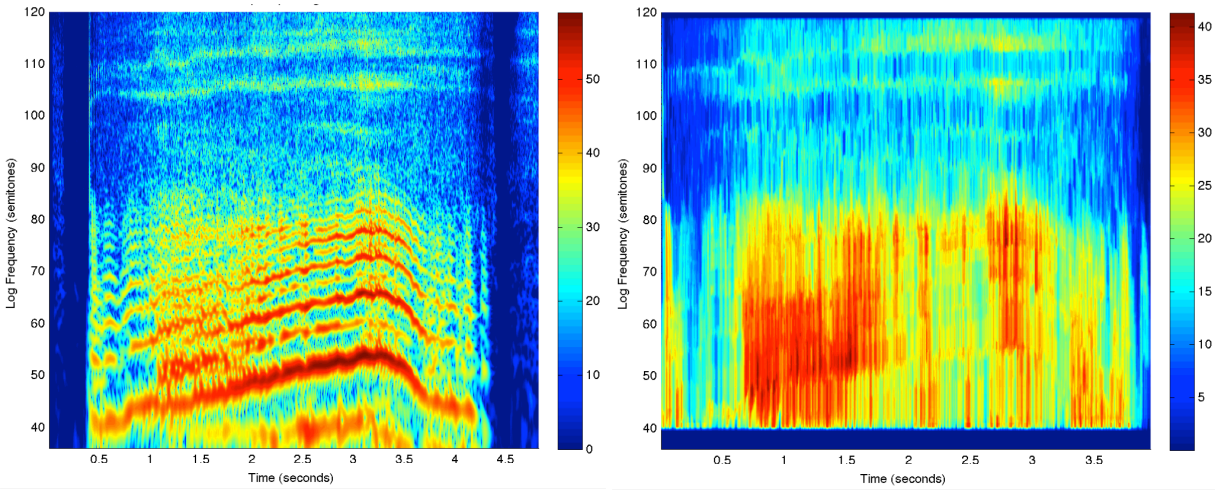


Figure 32: The original spectrogram (left) and the resulting removal of spectral peaks (right). The sound level scale expressed by the colour bar is normalized to the maximum value. Same sound as in Figure 1.

4.4 Prediction classes and methods

As mentioned above, the current prediction focussed on the three separate articulatory classes *phonation*, *turbulence*, and *myoelastic vibration*. The prediction of these classes (each one including several subcategories) was partly a completely new challenge with non-existent attempts in previous literature. This was in particular true for the sounds produced within the project using extended vocal technique. Each category can be active in a rather independent way, thus for example, myoelastic vibration using the tongue and lips can be combined with both phonation and turbulence. Therefore, for each category we made an independent classification/regression model. The ground truth was coded as 1 for the positive segments and 0 for the negative ones. See Table 1 for the distribution across classes and subjects.

Due to the relatively large number of features (54) in relation to the total number of cases (438, 434, 453, for each class, respectively) we used as the main prediction method Partial Least-Square Regression (PLS) in combination with cross-validation. PLS regression attempts to minimize the number of independent features by a principal component analysis in combination with a regression [GK86]. The method can be used when there are a large number of interdependent features. We used the built-in PLS package in Matlab for this computation.

For the classification of the positive and negative category, the regression data was simply categorised as true (1) for values higher than 0.5 and otherwise false (0). The number of

factors in the PLS regression was selected manually by choosing the minimum number that could still explain a major part of the cross-validated variation.

Two different cross-validation methods were used. The first one was the traditional 10-fold method with 100 random permutations. The second was "leave-one-subject-out". Since there were a total of 4 subjects in this study, the training was performed on three subjects and the testing on the remaining one. Then it was repeated for all subjects. This would correspond more closely to a real world case when a possible project prototype system is operated by a new user. However, due to the small number of subjects it is highly sensible to individual variations and therefore not so reliable as an estimate.

For comparison we also used standard Support Vector Machine (SVM) classification. Here we used the LIBSVM package for Matlab [CL11].

4.5 Results

4.5.1 Correlations with ground truth

As a first test Pearson's correlation coefficients were computed between each feature and the ground truth. Table 21, 22, 23 displays the correlations for all the phonation, myoelastic, and turbulence features, respectively. As seen in the Table 21, almost all features are significantly correlated with the ground truth. However, the highest correlations are found for the phonation class with correlations around $r = 0.7$ indicating that these features are to a certain part capturing some of the unique properties of the phonation class.

In Table 22 we see that the group of features extracted from spectrogram (vib...) reasonably correlates with the myoelastic ground truth although the maximum correlations in this case reaches about $r = 0.5$. For the group of features extracted from the time signal (cor_....) the correlations are in fact higher to the phonation category. This is somewhat expected since the autocorrelation method YIN was originally developed for pitch detection.

For the turbulence features shown in Figure 23, the correlations to the turbulence ground truth (rightmost column) is rather weak indicating that these features does not particularly well capture the intended variation.

4.5.2 Prediction of phonation

The results of the prediction of phonation for different cross-validations and methods are summarized in Table 24. As shown in the Table, all methods obtained an overall classification accuracy above 90 % using a modest number of PLS components. While the SVM method obtained the best classification for the ten-fold cross-validation (94.2 %), PLS obtained better results for the more challenging task of leave-one-subject-out cross-validation (92.0 %). Thus, the PLS method using 5 components seem to be a better choice for a future application when the model will be applied on new unknown data.

In Figure 33 the results from the PLS regression is shown when applied using 5 components and without cross-validation. Thus, this is the prediction output before the classification is performed. As seen in the figure, there are clearly two groups divided by the classification boundary at 0.5. Interestingly, the overall accuracy without cross-validation increased rather modestly to 94.1 % indicating that there is very little over-fitting using this method.

Feature	phonation (n=438)	myoelastic (n=434)	turbulence (n=453)
duration	0.20***	0.13**	-0.07
hf0_maxsl_uqt	0.73***	-0.26***	-0.40***
hf0_maxsl_std	0.69***	-0.20***	-0.36***
hf0_maxsl_mva	0.03	0.09	-0.07
hf0_maxf0_uqt	-0.66***	-0.23***	0.35***
hf0_maxf0_std	-0.56***	0.14**	0.27***
hf0_maxf0_mva	-0.64***	0.10*	0.29***
hf0_max2sl_uqt	0.68***	-0.22***	-0.43***
hf0_max2sl_std	0.60***	-0.11*	-0.37***
hf0_max2sl_mva	-0.27***	0.28***	0.05
hf0_max2f0_uqt	-0.52***	-0.10*	0.29***
hf0_max2f0_std	-0.40***	0.20***	0.22***
hf0_max2f0_mva	-0.70***	0.16***	0.33***
hf0_meanrestsl_uqt	0.61***	-0.23***	-0.41***
hf0_meanrestsl_std	0.59***	-0.18***	-0.39***
hf0_meanrestsl_mva	0.24***	0.14**	-0.25***
hf0_maxsldiff_uqt	0.72***	-0.26***	-0.37***
hf0_maxsldiff_std	0.72***	-0.18***	-0.34***
hf0_maxsldiff_mva	-0.21***	0.18***	0.12*

Table 21: Correlations between **phonation** features and ground truth for the three articulation classes.

Feature	phonation (n=438)	myoelastic (n=434)	turbulence (n=453)
vibrate1	-0.07	0.49***	-0.11*
vibext1	-0.10*	0.50***	-0.16***
vibprop1	-0.09*	0.51***	-0.15**
vibextprop1	-0.10*	0.47***	-0.18***
vibrate2	-0.01	0.46***	-0.13**
vibext2	-0.09	0.47***	-0.13**
vibprop2	-0.04	0.43***	-0.20***
vibextprop2	-0.05	0.17***	0.00
cor_med1	0.54***	-0.34***	-0.32***
cor_mva1	0.44***	-0.32***	-0.29***
cor_std1	0.53***	-0.32***	-0.33***
cor_med2	0.50***	-0.33***	-0.28***
cor_mva2	0.44***	-0.33***	-0.29***
cor_std2	0.52***	-0.32***	-0.32***
cor_diff	0.45***	-0.32***	-0.23***
cor_gaus	-0.55***	0.35***	0.34***

Table 22: Correlations between **myoelastic** features and ground truth for the three articulation classes. The vib... features are extracted from the spectrogram while the cor_... features are extracted from the time signal.

Feature	phonation (n=438)	myoelastic (n=434)	turbulence (n=453)
nosB2_1_uqt	-0.42***	0.46***	0.16***
nosB2_1_std	-0.09	-0.07	0.17***
nosB2_1_mva	-0.40***	0.31***	0.30***
nosB2_2_uqt	-0.53***	0.17***	0.32***
nosB2_2_std	-0.29***	0.12*	0.20***
nosB2_2_mva	-0.30***	0.49***	-0.00
nosB7_1_uqt	-0.44***	0.43***	0.10*
nosB7_1_std	-0.17***	0.02	0.19***
nosB7_1_mva	-0.45***	0.37***	0.30***
nosB7_2_uqt	-0.37***	0.45***	0.10*
nosB7_2_std	-0.11*	-0.05	0.16***
nosB7_2_mva	-0.43***	0.31***	0.30***
nosB7_3_uqt	-0.41***	0.45***	0.20***
nosB7_3_std	-0.05	-0.07	0.15**
nosB7_3_mva	-0.41***	0.28***	0.32***
nosB7_4_uqt	-0.49***	0.42***	0.23***
nosB7_4_std	-0.16**	0.04	0.20***
nosB7_4_mva	-0.47***	0.38***	0.28***
nosB7_5_uqt	-0.47***	0.36***	0.21***
nosB7_5_std	-0.27***	0.12*	0.20***
nosB7_5_mva	-0.40***	0.47***	0.12*
nosB7_6_uqt	-0.48***	0.12*	0.31***
nosB7_6_std	-0.27***	0.12*	0.18***
nosB7_6_mva	-0.32***	0.45***	0.03
nosB7_7_uqt	-0.61***	-0.09	0.46***
nosB7_7_std	-0.44***	-0.01	0.31***
nosB7_7_mva	-0.45***	0.38***	0.09

Table 23: Correlations between **turbulence** features and ground truth for the three articulation classes. nosB2_1 - nosB2_2 indicate the two frequency band division and nosB7_1 - nosB7_7 indicate the seven octave bands.

Cross-validation	PLS comp.	PLS accuracy (%)	SVM accuracy (%)
10-fold	6	93.8	94.2
Leave-one-subject-out	5	**92.0**	90.6

Table 24: Final prediction results for the **phonation** class.

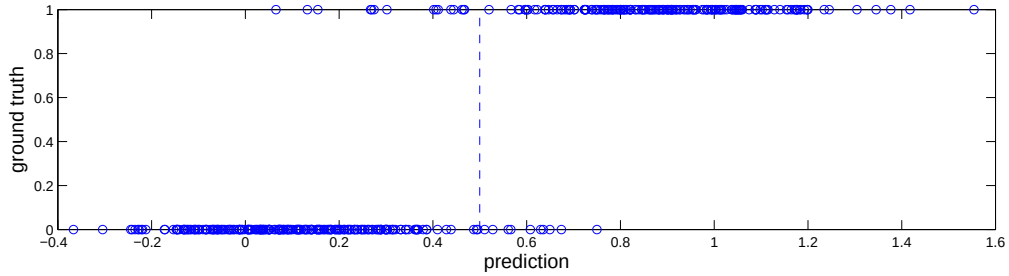


Figure 33: The output of the PLS regression using 5 components without cross-validation for the phonation model. The upper histogram shows the distribution of the prediction for ground truth = 1 (phonation) and the lower histogram for ground truth = 0 (no phonation). The dotted line marks the classification boundary.

4.5.3 Prediction of myoelastic vibrations

The results of the prediction of myoelastic vibrations for different cross-validations and methods are summarized in Table 25. Here the obtained overall classification accuracy was slightly lower than for the phonation class and ranged from 79 to 86 %. The PLS method obtained the best classification for both the ten-fold cross-validation (85.8 %), and leave-one-out cross-validation (84.6 %). Thus, the PLS method using 3 components seem to be the best choice.

In Figure 34 the results from the PLS regression is shown when applied using 3 components and without cross-validation. Here there is a larger overlap between the two groups. Also here the overall accuracy without cross-validation increased rather modestly to 87.1 % indicating that there is very little over-fitting using this method.

Cross-validation	PLS comp.	PLS accuracy (%)	SVM accuracy (%)
10-fold	3	85.8	84.0
Leave-one-subject-out	3	**84.6**	79.0

Table 25: Final prediction results for the **myoelastic** class.

4.5.4 Prediction of turbulence

The results of the prediction of turbulence for different cross-validations and methods are summarized in Table 26. Here the obtained overall classification accuracy was the lowest of the three classes and ranged from 76 to 81 %. The SVM method obtained slightly better classification for the ten-fold cross-validation (80.5 %), while the PLS method obtained the best results for leave-one-out cross-validation (77.7 %). Thus, the PLS method using 3 components seem to be the best choice.

In Figure 35 the results from the PLS regression is shown when applied using 3 components and without cross-validation. Similar to the myoelastic prediction (Figure 4) there is an overlap between the two groups. There seems also to be a larger spread in the positive group (ground truth=1) possibly indicating that the specifically developed features did not sufficiently catch the key properties of turbulence. Also here the overall accuracy without cross-validation

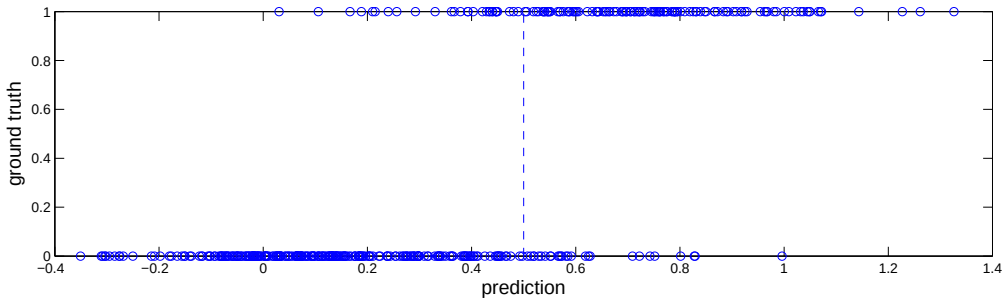


Figure 34: The output of the PLS regression using 3 components without cross-validation for the **myoelastic** model. The upper histogram shows the distribution of the prediction for ground truth = 1 (myoelastic) and the lower histogram for ground truth = 0 (not myoelastic). The dotted line marks the classification boundary.

increased rather modestly to 81.7 % indicating that there is very little over-fitting using this method.

Cross-validation	PLS comp.	PLS accuracy (%)	SVM accuracy (%)
10-fold	3	79.7	80.5
Leave-one-subject-out	3	**77.7**	75.7

Table 26: Final prediction results for the **turbulence** class.

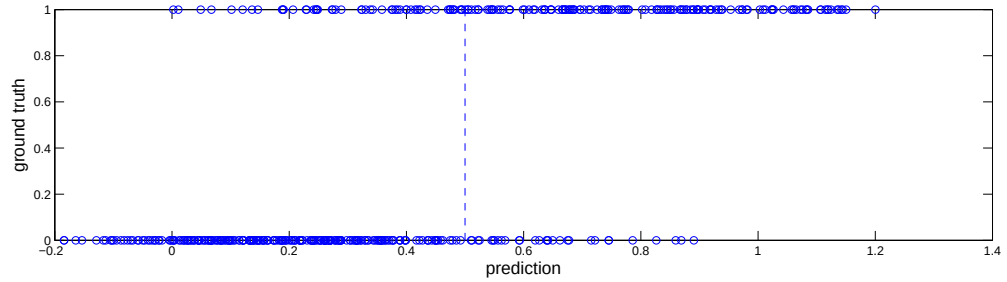


Figure 35: The output of the PLS regression using 3 components without cross-validation for the **turbulence** model. The upper histogram shows the distribution of the prediction for ground truth = 1 (turbulent) and the lower histogram for ground truth = 0 (not turbulent). The dotted line marks the classification boundary.

4.6 Summary and discussion

Using a set of features developed using extensions to the Auditory Receptive Fields toolbox the three different articulation classes phonation, myoelastic vibrations and turbulence were predicted using Partial Least-Square (PLS) regression and Support Vector Machines (SVM). The models that seem to have best generalizability were obtained by the PLS method using

either 3 or 5 components. The classification accuracy was then for 10-fold cross validation 93.8 % for phonation, 85.8 % for myoelastic, and 79.7 % for turbulence.

Possibly, the result for SVM could be further improved using feature selection methods and parameter optimization. This was not tested in the current study. One possible reason for the lower results for myoelastic and turbulence class could be the unbalanced groups - these classes contained proportionally more cases in the negative groups. This could possibly also be improved by further optimization of parameters both for the PLS and SVM method.

The lowest results were obtained for the turbulence class. We assumed a priori that turbulence should be strongly related to the amount of noise in the signal. Obviously, air turbulence will generate noise. However, the definition of turbulence in the annotations is a bit different. For example, myoelastic vibrations without any extra sound source are not classified as turbulent, although, there is a considerable portion of noise in the signal. Further comparison of the criteria for the annotations as well as an analysis of the incorrectly classified examples in each class seems to be an important path for future development.

References

[Add05] Paul S Addison. Wavelet transforms and the ECG: a review. *Physiological Measurement*, 26(5):R155, 2005.

[BD04] Y Boers and J N Driessen. Multitarget particle filter track before detect application. *Radar, Sonar and Navigation, IEE Proceedings*, 151(6):351–357, 2004.

[CH08] A. Camacho and J. G. Harris. A sawtooth waveform inspired pitch estimator for speech and music. *Journal of the Acoustical Society of America*, 124:1638–1652, 2008.

[CL11] Chih-Chung Chang and Chih-Jen Lin. Libsvm: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 2(3):27, 2011.

[CP14] Mark Cartwright and Bryan Pardo. Synthassist: Querying an audio synthesizer by vocal imitation. In *Proceedings of the International Conference on New Interfaces for Musical Expression*, pages 363–366, London, United Kingdom, 2014. Goldsmiths, University of London.

[DCK02] Alain De Cheveigné and Hideki Kawahara. Yin, a fundamental frequency estimator for speech and music. *The Journal of the Acoustical Society of America*, 111(4):1917–1930, 2002.

[EF15] Anders Elowsson and Anders Friberg. Modeling the perception of tempo. *The Journal of the Acoustical Society of America*, 137(6):3163–3177, 2015.

[EPS14] Samuel P. Ebenezer and Antonia Papandreou-Suppappola. Multiple transition mode multiple target track-before-detect with partitioned sampling. In *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 8008–8012. IEEE, May 2014.

[Fit10] Derry Fitzgerald. Harmonic/percussive separation using median filtering. In *Proc. of the 13th Int. Conference on Digital Audio Effects (DAFx-10)*, 2010.

[FSJ05] Anders Friberg, Erwin Schoonderwaldt, and Patrik N Juslin. Cuex: An algorithm for extracting expressive tone variables from audio recordings. *Acoustica united with Acta Acoustica*, 93:411–420, 2005.

[GK86] Paul Geladi and Bruce R Kowalski. Partial least-squares regression: a tutorial. *Analytica chimica acta*, 185:1–17, 1986.

[Ken04] Adam Kendon. *Gesture: Visible action as utterance*. Cambridge University Press, 2004.

[LF15a] Tony Lindeberg and Anders Friberg. Idealized computational models for auditory receptive fields. *PLOS ONE*, 10(3):e0119032–1, 2015.

[LF15b] Tony Lindeberg and Anders Friberg. Scale-space theory for auditory signals. In *Scale-Space and Variational Methods in Computer VisionSSVM 2015*, volume 9087, pages 3–15. Springer, 2015.

[Mal08] Stephane Mallat. *A Wavelet Tour of Signal Processing, Third Edition: The Sparse Way*. Academic Press, 3rd edition, 2008.

[McC74] S. McCandless. An algorithm for automatic formant extraction using linear prediction spectra. *Acoustics, Speech and Signal Processing, IEEE Transactions on*, 22(2):135–141, Apr 1974.

[MG76] J. D. Markel and A. H. Gray. *Linear Prediction of Speech*. Springer-Verlag, 1976.

[MP] E. Marchetto and G. Peeters. A set of audio features for the morphological description of vocal imitations. In *Proc. of 18th Int'l Conference on Digital Audio Effects (DAFx-15)*.

[MW05] P. McLeod and G. Wyvill. A smarter way to find pitch. In *Proc. of Int'l Computer Music Conference*, 2005.

[NR] V. Niennattrakul and C. A. Ratanamahatana. Shape averaging under time warping. In *Proc. of Int'l Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology*.

[Pee04] Geoffroy Peeters. A large set of audio features for sound description (similarity and classification) in the Cuidado project. Cuidado project report, IRCAM, 2004.

[Pra94] Eric Prame. Measurements of the vibrato rate of ten singers. *The journal of the Acoustical Society of America*, 96(4):1979–1984, 1994.

[QZLL15] Changzhen Qiu, Zhiyong Zhang, Huanzhang Lu, and Huiwu Luo. A Survey of Motion-Based Multitarget Tracking Methods. *Progress In Electromagnetics Research B*, 62:195–223, 2015.

[Rao11] Vishweshwara Mohan Rao. *Vocal Melody Extraction from Polyphonic Audio with Pitched Accompaniment*. PhD thesis, Indian Institute of Technology Bombay, 2011.

[RBS+11] Nicolas Rasamimanana, Frédéric Bevilacqua, Norbert Schnell, Emmanuel Fléty, and Bruno Zamborlin. Modular Musical Objects Towards Embodied Control Of Digital Music Real Time Musical Interactions. In *Proceedings of the fifth international conference on Tangible, embedded, and embodied interaction*, TEI'11, pages 9–12, Funchal, Portugal, 2011.

[RH07] A Rosenberg and J Hirschberg. V-Measure: A Conditional Entropy-Based External Cluster Evaluation Measure. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 410–420, 2007.

[SB01] D J Salmond and H Birch. A particle filter for track-before-detect. In *Proceedings of the American control conference*, volume 5, pages 3755–3760, 2001.

[SJB14] J. Salamon, C. Jacoby, and J. P. Bello. A dataset and taxonomy for urban sound research. In *22st ACM International Conference on Multimedia (ACM-MM'14)*, Orlando, FL, USA, Nov. 2014.

[SRS+09] Norbert Schnell, Axel Röbel, Diemo Schwarz, Geoffroy Peeters, and Riccardo Borghesi. Mubu & friends - assembling tools for content based real-time interactive audio processing in max/msp. In *Proceedings of International Computer Music Conference*, Montreal, 2009.

[TC98] Christopher Torrence and Gilbert P Compo. A practical guide to wavelet analysis. *Bulletin of the American Meteorological society*, 79(1):61–78, 1998.

FP7-ICT-2013-C FET-Future Emerging Technologies-618067

SkAT-VG: Sketching Audio Technologies using Vocalizations and Gestures

D5.5.1 Extension
Extended results for audio recognition of vocal primitives (KTH)

First Author	Anders Friberg
Responsible Partner	IRCAM
Status-Version:	Final-1.0
Date:	December 15, 2016
EC Distribution:	Consortium
Project Number:	618067
Project Title:	Sketching Audio Technologies using Vocalizations and Gestures

Title of Deliverable:	Extended results for audio recognition of vocal primitives (KTH)
Date of delivery to the EC:	14/12/2016

Workpackage responsible for the Deliverable	WP5
Editor(s):	Anders Friberg
Contributor(s):	Anders Friberg, Petúr Helgason, Anders Elowsson
Reviewer(s):	
Approved by:	All Partners

Abstract	
Keyword List:	

Document Revision History

Version	Date	Description	Author
First version	14/12/2016		AF

Table of Contents

Index of Figures

List of Acronyms and Abbreviations

DoW Description of Work

EC European Commission

PM Person Months

WP Work Package

1 Summary

This report presents additional work extending the section about audio recognition of vocal primitives that was presented in delivery D5.5.1. The previous method has been improved regarding the database, the annotation, the feature computation, and the machine learning methods. The database of the previous four Swedish speakers was extended with four French speakers thus doubling the total size. The extraction of annotation labels for the extracted sounds were modified to also including subcategories of the three main categories phonation, myoelastic vibrations and turbulence. Several improvements were introduced for the myoelastic and turbulence features. In particular, we developed of a new set of features for better capturing the myoelastic vibrations in the vocal imitations by means of a modulation filter bank. These new features were found to better capture the myoelastic vibrations improving both the prediction of the myoelastic class as well as the prediction of the other classes. Three different machine learning methods were tested for predicting the final articulatory categories. In addition to the previously used methods (PLS and SVM) an ensemble of neural networks (ENNs) was also included. The result with the best generalization was obtained using the ENN. The resulting classification accuracy using 10-fold cross validation was 96.9 % for phonation, 90.9 % for myoelastic vibrations, and 88.5 % for turbulence. This was an improvement with 2.7 to 8.0 % in comparison with the previous method. The modulation filter bank alone was found to predict myoelastic vibrations with 84 % accuracy using only six bands and the SVM method.

2 Data set

The database used in this study consists of vocal imitations by four Swedish and four French speakers that were previously annotated regarding the detailed articulacy production. The collection of the data was described in D2.2.2 and the annotations were described in D3.3.2. We will here just present the representation of the dataset used in this study in which a mapping was done from the original annotations to three main categories as well as their subcategories as described in the following section.

2.1 Annotations

As in the previous prediction, described in D.5.5.1, we focused the work on the three main categories

- The presence vs. absence of *vocal fold phonation*.
- The presence vs. absence of slow *myoelastic vibration*.
- Third, the presence or absence of *turbulence* in the signal.

In addition, for each category, a number of subcategories were identified and marked according to the numbering below. This was primarily used for the development of the different features (in particular the myoelastic category). The reason was that each subcategory in some cases exhibited rather different acoustical results. For example, the frequency of

the myoelastic vibration varies substantially between different subcategories. Another reason was to make the dataset more general and flexible so that other recognition tasks could be performed on the same dataset.

Myoelastic
0 = no myoelastic present
1 = lax labial myoelastic vibration
2 = lax tongue tip myoelastic vibration
3 = lax uvular myoelastic vibration
4 = aryepiglottal/ventricular vibration
5 = velic myoelastic vibration
6 = tense labial myoelastic vibration
7 = tense uvular/pharyngeal vibration

All values from 1 to 5 are lax vibrations. Values 1-3, 5 can be expected to vibrate at 20-50 Hz. Value 4 will tend to be faster, at around 60-110 Hz, depending on the speaker (higher values for females). Values 6 and 7 produce much higher frequencies than the lax vibrations, 150-700 Hz.

Phonated
0 = no phonation
1 = breathy voice
2 = falsetto
3 = modal voice
4 = pressed voice
5 = creaky voice
6 = unspecified vocal fold phonation

The values 1 to 4 should all yield periodicity, but 5 will likely be aperiodic. Value 6 is used in cases when the vocal folds are vibrating, but not producing a "normal" voice. Often it's like the voice is missing the fundamental and vibrating in the second or third mode. This sometimes happens at the onset or offset of voicing, for example, as if the vocal folds fail to engage properly.

Turbulent
0 = no turbulence
1 = labial turbulence
2 = turbulence with a grooved tongue anterior (s- and sh-like turbulence)
3 = turbulence with a flat tongue anterior (th- and x-like turbulence)
4 = turbulence with a lateral tongue anterior opening (lateral turbulence)
5 = glottal turbulence (h-like sound)
6 = nasal turbulence
7 = turbulence due to a myoelastic constriction

There should be systematic spectral differences between these groups. Values 1 and 6 should yield fairly flat spectra, since there is not much of a cavity anterior to the source. Value 2 should yield the highest frequency peaks. Value 3 will vary from a flat spectrum (th-like sounds) to fairly high frequency palatal friction, to low frequency (as fricatives go) uvular friction. Value 5 can be expected to vary, because the filtering depends on the adjacent context. Value 7 is a special case, which mostly has to do with how uvular fricatives tend to be produced. They tend to have a myoelastic constriction that produces a harsh type of turbulence (rather than the channel turbulence one sees in, e.g., palatal fricatives). It would be possible to separate anterior and posterior friction and create 2 categories out of value 3.The posterior fricatives are associated with lower resonances than the anterior ones.

2.2 Final extraction

The procedure described above generated a list of segment data pointers into the original database files. The final audio excerpts were extracted using a script in Matlab resulting in one audio file for each example. Each file name was marked with the three-number combination described above. Each excerpt contained only one type and subtype of articulation. Thus, we limit this study to examples that are already segmented.

The analysis of in particular slow myoelastic vibrations demands a certain time window. The duration limit of the included segments was therefore set to 150 ms. This corresponds to three cycles of 20 Hz which is approximately the lower frequency bound for myoelastic vibrations. This made also the balance between the number of segments in the positive and negative categories in each class more even.

The extraction resulted in a total of 2689 audio segments of which 1242 were longer than 150 ms. and thus kept for the modeling. The final distribution of the three data sets is shown in Table 1. There is a reasonable even distribution of the number of segments in the positive and negative groups for phonation and turbulence. There are comparatively fewer myoelastic cases, which result in a larger portion of negative segments in the myoelastic class. The number of segments varies across speakers with comparatively more segments for the French speakers.

3 Features

A group of features was previously developed specifically for each targeted articulation class as specified in D5.5.1, section 4.3. We will here just describe the changes relative that description. The turbulence features were changed and a new set of features for the myoelastic vibrations were added as described below. The rest of the features that are described in D5.5.1 were kept for the subsequent classification.

3.1 Turbulence features

The extraction of turbulence features was further refined and extended in several ways. Therefore the full description is given here. The spectral shape of the noise part was calculated as before but with a refined smoothing in frequency. Also, the spectral shape given

	Gender	Nationality	Total	Phonation	Myoelastic	Turbulence
Pos. Total				698	300	705
Neg. Total				543	941	536
Speaker 1	M	Sw.	143	73/70	48/95	78/65
Speaker 2	M	Sw.	81	45/36	23/58	41/40
Speaker 3	F	Sw.	110	53/57	35/75	49/61
Speaker 4	F	Sw.	87	47/40	62/144	43/44
Speaker 5	M	Fr.	206	120/86	36/152	128/78
Speaker 6	M	Fr.	188	115/70	48/70	72/116
Speaker 7	F	Fr.	223	127/96	10/213	183/40
Speaker 8	F	Fr.	203	118/85	57/146	111/92

Table 1: The distribution of the number of segments in each of the three articulation classes. Numbers for each subject refer to the number of positive/negative segments.

by the spectral peaks was estimated. A new measure of harmonics-to-noise was defined and used as a feature. For the turbulence category we applied the following steps to the resulting spectrogram calculated in step two in D5.5.1, section 4.1 Phonation features:

1. Smoothing in time

The spectrogram was smoothed in time using a receptive field applied on the spectrogram. A discrete Gaussian kernel was used with a standard deviation of 30 ms.

2. Estimation of noise spectrum using a smoothing filter

A smoothing filter was specifically designed to remove the harmonic partials considering the bandwidth of the spectrogram and the variation of the harmonic density across the spectrum. A filter with a varying window as a function of frequency was applied using the 15 % percentile. It can be viewed as a development of median filtering used for separating harmonic and percussive content ([Fit10]; [EF15]). The filter window varied according to a linearly interpolated break-point function as a function of frequency. The obtained spectral shape was further smoothed in the frequency domain using a similar frequency-dependent filter using a Gaussian kernel. The percentile and the break-points was chosen manually using spectrum displays in order to minimize the harmonic content while retaining most of the turbulence (noise).

3. Estimation of spectral peaks using a smoothing filter The spectral shape following the peaks in the spectrum was estimated using the same filter and smoothing as in step 2 above. The difference is that this filter was using the 95 % percentile instead.

An illustration of the resulting estimations of noise and spectral peaks at a specific time is shown in Figure 1 and for the whole spectrogram in Figure 2.

4. Frame feature extraction (spectral bands) The remaining spectral shape of the noise in step 2 is divided both into seven octave bands (with boundaries at 36, 48, 60 72, 84, 96, 108, 120 semitones in MIDI units) and into two bands (above or below 1kHz). For each band (and frame) the median value across frequency was calculated.

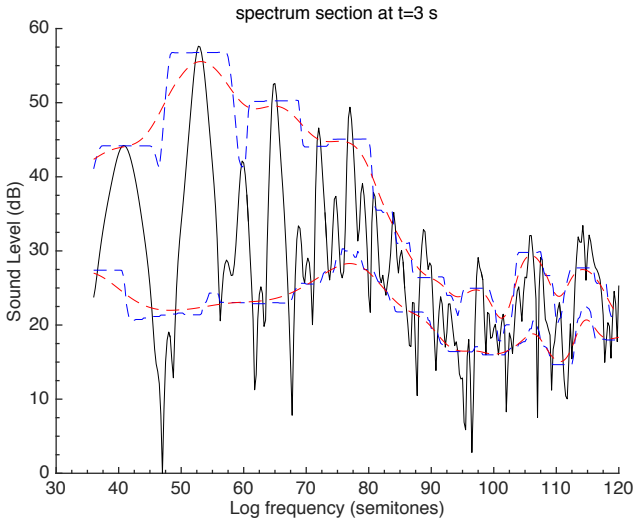


Figure 1: The smoothing filters with variable windows applied on a spectrum section at t = 3 s. The black line indicates the original spectrum, the dashed blue lines are the percentile filtered spectra, and the dashed red lines are the resulting spectra after smoothing for the upper and lower estimation. The sound is an imitation of an accelerating lorry containing both phonation and turbulence and the same example as used in D5.5.1 (Figure 29).

5. Harmonics-to-noise ratio

Using the upper and lower estimation of the spectra above, a simple measure of harmonics-to-noise is computed by taking the maximum distance between the two curves.

6. Final features across frames

The final features are calculated in the same way as for the phonation features. Thus using the upper quartile, standard deviation, and the mean of the difference between frames (mean derivative in time). This resulted in a total number of 28 (7 bands * 3 stats + 2 bands * 3 stats + 1 ratio) features for the turbulence class.

3.2 Modulation filter bank added to the myoelastic features

The myoelastic sounds can be viewed as a comparatively slow amplitude modulation of the turbulence and/or phonation. Therefore, we developed a modulation filter bank for specifically detecting AM modulation in the lower frequency range between 20 and 1000 Hz. The design can be viewed as a simplification of the auditory modulation filter bank proposed by Dau et al. [DKK97]. The following steps describe the procedure.

1. Instant sound level

The instant sound level SL was computed using a Hanning window of 1ms and a hop size of 0.1 ms. A highpass filter with a cutoff at 15 Hz removed any DC component. Thus, the resulting SL had a sampling frequency of 10 kHz.

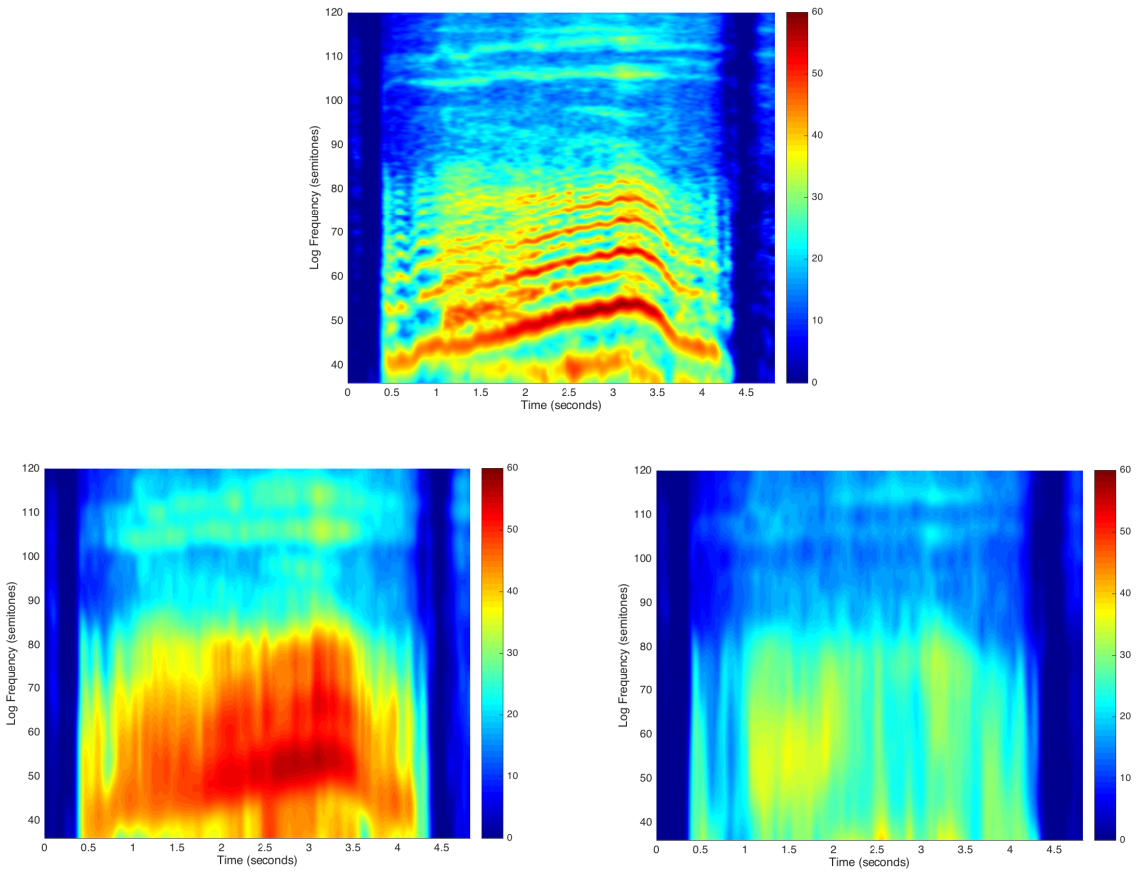


Figure 2: Resulting spectral shapes in the turbulence feature extraction. The smoothed spectrogram (top), the spectral shape of the peaks (left) and the spectral shape of the noise after removal of narrow spectral peaks (right). Same sound as in previous figure.

2. Spectrum

An average spectra was computed by averaging over a series of FFTs with a Hamming window size of 1024 samples and an overlap of 512 samples.

3. Filter bank

Six filter channels were defined according to the expected frequencies for the different myoelastic vibrations as specified in section 1.1 above. They were approximately logarithmically distributed with band 1, 18-35 Hz; band 2, 35-60 Hz; band 3, 60-110 Hz; band 4, 110-220 Hz; band 5, 220-500 Hz; band 6, 500 -1000 Hz.

4. Final features

The maximum sound levels in each band were used as the final six features.

3.3 Myoelastic features using vibrato extraction methods on the sound level waveform

Here we used an alternative extraction of the main AM frequencies applying a method developed for vibrato extraction by Friberg et al. [FSJ05]. The instant SL curve as described above was used as input. The SL curve was first filtered and peaks were detected using the vibrato extraction method. This was done in three different frequency areas. The following steps were applied:

1. Filtering

The SL signal was lowpass and highpass filtered into three different bands corresponding to the expected frequencies for myoelastic vibrations. Band 1, 15-40 Hz; band 2, 40-110 Hz; band 3, 100-250 Hz.

2. Detect regular amplitude variations

In each band, cyclic variations of the SL curve were detected using the three-point method suggested by Prame [Pra94] and implemented by Erwin Schoonderwaldt [FSJ05]. The peaks and troughs of the band filtered SL curve were detected with a simple peak-picking method. For each half-cycle (peak-trough-peak or trough-peak-trough) rate R and extent E were calculated using a three-point estimation involving the two adjacent peaks/troughs:

$$R_n = \frac{1}{t_{n+1} - t_{n-1}},$$

$$E_n = \frac{|A_{n+1} - 2A_n + A_{n-1}|}{4},$$

where tn indicates the time instance of the peak/trough and A is the filtered SL.

3. Final features

There were four features calculated from the detected points for each band. The first two features consisted of the median over all detected points for the rate R and extent E. The third feature was a calculation of the total length of detected variations relative to the total length

of the example. The fourth feature was the multiplication of the median extent feature (No. 2) with the relative duration feature (No. 3). In addition, two combinations of the extent values were computed using the maximum across band 1 and 2 and the maximum across band 1, 2, and 3. This resulted in a total of 4*3+2=14 features.

4 Prediction classes and methods

As described before (D5.5.1 section 4.4) we used partial-least square regression (PLS) and support vector machine classification (SVM). In addition, we used also an ensemble of neural networks (ENN). As each neural network (NN) was randomly initialized, they will converge at different local minima. The ensemble of these networks will therefore act as a regularization technique that enhances generalization capabilities [HS90]. In other words, when using the average prediction of these models, we can expect a better outcome than if we were to randomly choose one of them [Pol06]. After initial testing, the following setup was chosen for each NN of the ensemble:

- Each network had three hidden layers, with 15 neurons in each layer. This resulted in an architecture (including input and output layer) of 84, 15, 15, 15, 1. Each network was thus rather deep, although the number of neurons was still kept small.
- The non-linearities in the first two hidden layers were hyperbolic tangent (tanh) units, and the non-linearities for the last hidden layer were rectified linear units (ReLUs). The idea of a mixture of non-linearities within an ensemble of neural network was introduced by [Elo16]. The output layer had a sigmoid activation function.
- The networks were trained with scaled conjugate gradient backpropagation.
- Each network was trained for a maximum of 240 epochs. Training was however set to stop if the gradient reached below 10-6.
- Each input feature was normalized to fit the range -1 to 1.
- We used all features in each model.

The 10-fold cross-validation was repeated 20 times (Monte-Carlo repetitions) for each method to ensure the consistency of our results.

5 Results

5.1 Correlations with ground truth

As a first test the point-biserial correlation coefficients were computed between each feature and the ground truth. Table 2, 3, 4 displays the correlations for all the phonation, myoelastic, and turbulence features, respectively. Due to the multiple testing the significance values should be interpreted with some caution and should only be viewed as an overall indication of the correspondence. As seen in Table 2, 3, 4 almost all features are significantly correlated with

the ground truth. However, the highest correlations are found for the phonation class with correlations up to $r_{pb} = 0.8$ indicating that these features are to a certain part capturing some of the unique properties of the phonation class.

Feature	Phonation	Myoelastic	Turbulence
hf0_maxsl_uqt	0.80***	-0.09**	-0.49***
hf0_maxsl_std	0.74***	-0.05	-0.17***
hf0_maxsl_mva	0.08**	0.09**	0.44***
hf0_maxf0_uqt	-0.74***	-0.23***	0.37***
hf0_maxf0_std	-0.73***	-0.01	0.41***
hf0_maxf0_mva	-0.76***	-0.08**	-0.55***
hf0_max2sl_uqt	0.77***	-0.05	-0.51***
hf0_max2sl_std	0.66***	0.01	0.06*
hf0_max2sl_mva	-0.36***	0.19***	0.27***
hf0_max2f0_uqt	-0.68***	-0.12***	0.24***
hf0_max2f0_std	-0.56***	0.06*	0.44***
hf0_max2f0_mva	-0.81***	-0.04	-0.53***
hf0_meanrestsl_uqt	0.63***	-0.05	-0.50***
hf0_meanrestsl_std	0.61***	-0.03	-0.29***
hf0_meanrestsl_mva	0.48***	0.21***	-0.47***
hf0_maxsldiff_uqt	0.78***	-0.10***	-0.48***
hf0_maxsldiff_std	0.75***	-0.01	0.10***
hf0_maxsldiff_mva	-0.25***	0.16***	-0.51***

Table 2: Correlations between phonation features and ground truth for the three articulation classes. Significance levels *$p < 0.05$, **$p < 0.01$, ***$p < 0.001$.

In Table 3 we see that the group of features extracted from spectrogram (vib...1-2) reasonably correlates with the myoelastic ground truth although the maximum correlations in this case reaches only about $r_{pb} = 0.28$. The new filter bank features (vibspecdb1-6) and the vibrato extraction methods on the SL (vib...3-5) correlate positively for lower bands with myoelastic vibration and negatively with phonation as expected. For the autocorrelation features extracted from the time signal (cor...) the correlations are in fact higher to the phonation category. This is somewhat expected since the autocorrelation method YIN was originally developed for pitch detection.

For the turbulence features shown in Figure 4, the correlations to the turbulence ground truth (rightmost column) vary considerably and reaches $r_{pb} = 0.58$ for the highest frequency band (nosB7_7_uqt). The new harmonics to noise measure nosH2N... correlates strongly with phonation indicating that it is capturing the intended information.

5.2 Prediction of phonation

The results of the prediction of phonation for different cross-validations and methods are summarized in Table 5. As shown in the table, all methods obtained an overall classification accuracy above 95 % for both cross-validations. The best PLS results were obtained with

feature	phonation	myoelastic	turbulence
vibrate1	-0.28***	0.21***	0.12***
vibext1	-0.28***	0.25***	0.05
vibprop1	-0.26***	0.26***	0.03
vibextprop1	-0.22***	0.28***	-0.03
vibrate2	-0.26***	0.17***	0.19***
vibext2	-0.28***	0.23***	0.11***
vibprop2	-0.24***	0.22***	0.06*
vibextprop2	-0.22***	0.10***	0.20***
vibspecdb1	-0.52***	0.21***	0.21***
vibspecdb2	-0.52***	0.37***	0.25***
vibspecdb3	-0.44***	0.39***	0.20***
vibspecdb4	-0.20***	0.27***	0.13***
vibspecdb5	0.42***	0.20***	-0.23***
vibspecdb6	0.65***	0.10***	-0.42***
cor_med1	0.60***	-0.16***	-0.42***
cor_mva1	0.58***	-0.09***	-0.42***
cor_std1	0.69***	-0.11***	-0.46***
cor_med2	0.53***	-0.17***	-0.37***
cor_mva2	0.56***	-0.15***	-0.40***
cor_std2	0.65***	-0.15***	-0.45***
cor_diff	0.46***	-0.18***	-0.32***
cor_gaus	-0.63***	0.13***	0.42***
vibrate3	-0.39***	0.23***	0.17***
vibext3	-0.32***	0.11***	0.12***
vibprop3	-0.44***	0.15***	0.20***
vibextprop3	-0.34***	0.12***	0.13***
vibrate4	-0.43***	0.24***	0.18***
vibext4	-0.33***	0.32***	0.08**
vibprop4	-0.44***	0.32***	0.17***
vibextprop4	-0.33***	0.37***	0.06*
vibrate5	-0.22***	0.11***	0.23***
vibext5	0.01	0.24***	-0.02
vibprop5	-0.29***	0.23***	0.11***
vibextprop5	-0.03	0.27***	-0.07*
vibcomb34	-0.29***	0.27***	0.05
vibcomb345	-0.04	0.21***	-0.03

Table 3: Correlations between myoelastic features and ground truth for the three articulation classes. The vib...1-2 features are extracted from the spectrogram using the vibrato detection method, the vibspecdb1-6 are the new modulation filter bank values, the vib...3-5 are extracted from sound level using vibrato method, and the cor... features are extracted from the time signal. Significance levels *$p < 0.05$, **$p < 0.01$, ***$p < 0.001$.

feature	phonation	myoelastic	turbulence
nosB2_1_uqt	-0.51***	0.28***	0.27***
nosB2_1_std	-0.11***	-0.08**	0.07*
nosB2_1_mva	-0.23***	0.21***	0.22***
nosB2_2_uqt	-0.75***	-0.02	0.50***
nosB2_2_std	-0.49***	0.03	0.23***
nosB2_2_mva	-0.29***	0.30***	0.07*
nosB7_1_uqt	-0.52***	0.24***	0.27***
nosB7_1_std	-0.18***	-0.04	0.15***
nosB7_1_mva	-0.34***	0.28***	0.30***
nosB7_2_uqt	-0.44***	0.29***	0.21***
nosB7_2_std	-0.10***	-0.08**	0.10***
nosB7_2_mva	-0.26***	0.19***	0.22***
nosB7_3_uqt	-0.49***	0.30***	0.26***
nosB7_3_std	-0.10***	-0.08**	0.07*
nosB7_3_mva	-0.18***	0.14***	0.19***
nosB7_4_uqt	-0.61***	0.21***	0.38***
nosB7_4_std	-0.30***	0.03	0.15***
nosB7_4_mva	-0.29***	0.26***	0.23***
nosB7_5_uqt	-0.66***	0.11***	0.41***
nosB7_5_std	-0.43***	0.05	0.20***
nosB7_5_mva	-0.30***	0.26***	0.17***
nosB7_6_uqt	-0.69***	-0.05	0.42***
nosB7_6_std	-0.44***	0.03	0.17***
nosB7_6_mva	-0.30***	0.27***	0.07**
nosB7_7_uqt	-0.76***	-0.16***	0.58***
nosB7_7_std	-0.64***	-0.06*	0.41***
nosB7_7_mva	-0.49***	0.16***	0.36***
nosH2N_uqt	0.81***	-0.10***	-0.48***
nosH2N_std	0.68***	-0.10***	-0.42***
nosH2N_mva	-0.82***	-0.11***	0.45***

Table 4: Correlations between turbulence features and ground truth for the three articulation classes. nosB2_1... - nosB2_2... indicate the two frequency band division, nosB7_1... - nosB7_7... indicate the seven octave bands, and nosH2N... is the new harmonics to noise measure. Significance levels *$p < 0.05$, **$p < 0.01$, ***$p < 0.001$.

a modest number of PLS components (5-6). The differences between methods were rather small. This indicates that the features were to a large extent able to capture the relevant acoustic properties for phonation versus non-phonation. The numbers in bold indicate the best method and the numbers in parenthesis indicate the improvement from the first model and testing reported in D.5.5.1. Note that these differences can be due to differences in the database (the number of subjects were doubled), the introduction of the features, and the introduction of the ENN method.

cross-validation	PLS comp.	PLS accuracy (%)	SVM accuracy (%)	ENN accuracy (%)
10-fold	6	96.4	96.6	96.9 (+2.7)
leave-one-subject-out	5	96.1 (+4.1)	95.4	95.9

Table 5: Final prediction results for the phonation class. The numbers in parenthesis are the improvements from the previous model.

In Figure 3 the results from the PLS regression is shown when applied using 6 components and without cross-validation. Thus, this is the prediction output before the classification is performed. As seen in the figure, there are clearly two groups divided by the classification boundary at 0.5. Interestingly, the overall accuracy without cross-validation increased rather modestly to 96.8 % indicating a small amount over-fitting using this method.

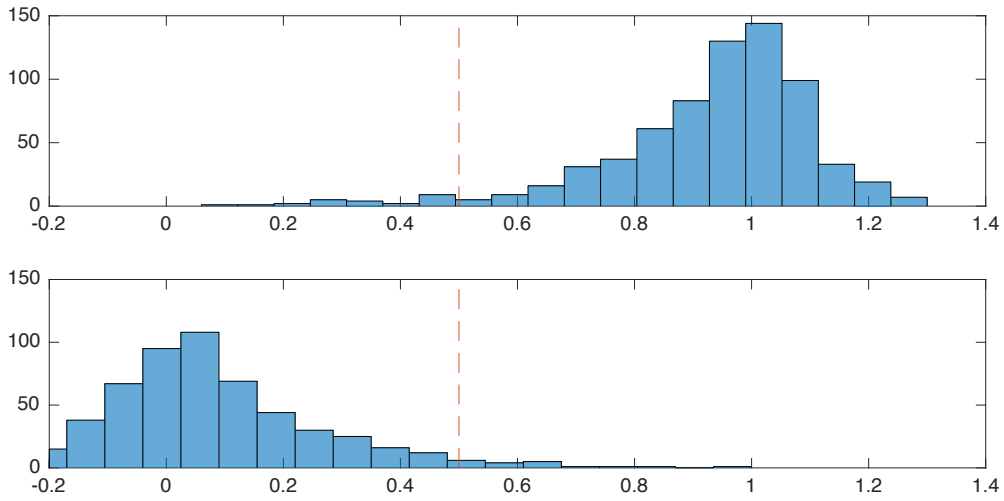


Figure 3: The output of the PLS regression using 5 components without cross-validation for the phonation model. The upper histogram shows the distribution of the prediction for ground truth = 1 (phonation) and the lower histogram for ground truth = 0 (no phonation).The dotted line marks the classification boundary.

5.3 Prediction of myoelastic vibrations

The results of the prediction of myoelastic vibrations for different cross-validations and methods are summarized in Table 6. Here the obtained overall classification accuracy was slightly lower than for the phonation class and ranged from 82 to 91 %. The ENN method obtained the best results and the differences between methods were larger. This indicates that the features still had some problems capturing the myoelastic acoustical properties. This is not so surprising since the myoelastic vibrations have a large span of different types each with different characteristic frequencies as listed in section 2.1 above.

cross-validation	PLS comp.	PLS accuracy (%)	SVM accuracy (%)	ENN accuracy (%)
10-fold	5	85.8	89.0	90.9 (+5.1)
leave-one-out	5	83.8	82.0	87.2 (+2.6)

Table 6: Final prediction results for the myoelastic class. The numbers in parenthesis are the improvements from the previous model.

In Figure 4 the results from the PLS regression is shown when applied using 5 components and without cross-validation. Here there is a larger overlap between the two groups. Also here the overall accuracy without cross-validation increased rather modestly to 86.5 % indicating a small amount of over-fitting using this method.

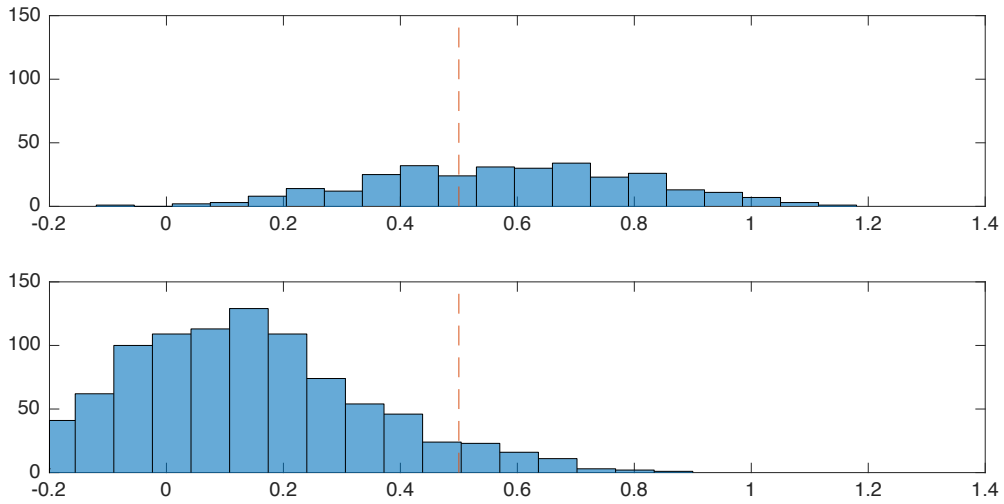


Figure 4: The output of the PLS regression using 5 components without cross-validation for the myoelastic model. The upper histogram shows the distribution of the prediction for ground truth = 1 (myoelastic) and the lower histogram for ground truth = 0 (not myoelastic). The dotted line marks the classification boundary.

For comparison we also evaluated the new modulation filter bank features separately. The PLS and SVM method was used to predict the myoelastic category using solely the 6 filter bank features. The result is presented in Table 7. The best results for 10-fold cross-validation

was obtained for SVM with an accuracy of 84 %. Compared to the to the SVM method for the full features set (Table 6) the decrease in accuracy was 5 % indicating that the filter bank features indeed capture some of the salient information within the whole feature set. Thus, a simplified model for myoelastic vibrations can be implemented using a set of well-known straightforward methods including sound level and FFT computations.

cross-validation	PLS comp.	PLS accuracy (%)	SVM accuracy (%)
10-fold	4	81.1	84.0
leave-one-out	4	80.0	80.7

Table 7: Final prediction results for the myoelastic class using only the 6 modulation spectrum bands (vibspecdb1-6).

5.4 Prediction of turbulence

The results of the prediction of turbulence for different cross-validations and methods are summarized in Table 8. Here the obtained overall classification accuracy was slightly lower than the myoelastic class and ranged from 78 to 89 %. The ENN method obtained the best results and there were noticeable differences between methods. Thus, it seems to perform in a similar way to the myoelastic class. This indicates that the features still had some problems capturing the turbulence acoustical properties. Contrary to the myoelastic features, the turbulence features were rather straightforward to extract and the intuitive impression was that they worked well. The rather low performance could possibly be attributed to the acoustic overlap between myoelastic and turbulence features. Turbulence is simply the existence of noise in the signal. However, also all different types of myoelastic vibrations generate noise although it is not annotated as turbulence. Thus, it is dependent on the detection of the myoelastic class to sort out these cases from the turbulence class by an interaction between all features. This also explains why there is a larger difference between methods. Note, that the PLS method is using a linear combination of features and thus do not include any interaction in the model.

cross-validation	PLS comp.	PLS accuracy (%)	SVM accuracy (%)	ENN accuracy (%)
10-fold	6	84.6	86.2	88.5 (+8.0)
leave-one-out	6	81.6	78.4	83.1 (+5.4)

Table 8: Final prediction results for the turbulence class. The numbers in parenthesis are the improvements from the previous model.

In Figure 5 the results from the PLS regression is shown when applied using 6 components and without cross-validation. Although the overall accuracy is comparable to the myoelastic case, the distribution indicates a better discrimination between the positive and negative groups in the figure. Also here the overall accuracy without cross-validation increased rather modestly to 86.1 % indicating a small amount of over-fitting using this method.

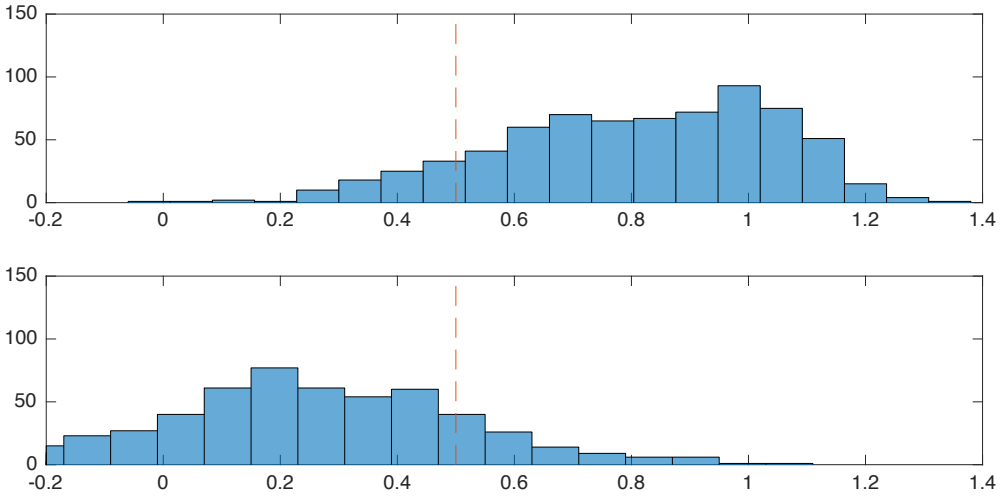


Figure 5: The output of the PLS regression using 6 components without cross-validation for the turbulence model. The upper histogram shows the distribution of the prediction for ground truth = 1 (turbulent) and the lower histogram for ground truth = 0 (not turbulent). The dotted line marks the classification boundary.

6 Summary

Using a set of features developed using extensions to the auditory receptive fields toolbox the three different articulation classes phonation, myoelastic vibrations and turbulence were predicted using Partial Least-Square (PLS) regression, Support Vector Machines (SVM), and a ensemble of neural networks (ENN).

The model that had the best performance was the ENN and the classification accuracy was for 10-fold cross validation 96.9 % for phonation, 90.9 % for myoelastic, and 88.5 % for turbulence.

The introduced features using a modulation filter bank were able to predict myoelastic vibration with an accuracy of 84 % (10-fold cross-validation) using only six features. Thus, it can be a starting point for making a rather simple implementation of the model using only standard signal processing techniques.

References

[DKK97] Torsten Dau, Birger Kollmeier, and Armin Kohlrausch. Modeling auditory processing of amplitude modulation. i. detection and masking with narrow-band carriers. *The Journal of the Acoustical Society of America*, 102(5):2892–2905, 1997.

[EF15] Anders Elowsson and Anders Friberg. Modeling the perception of tempo. *The Journal of the Acoustical Society of America*, 137(6):3163–3177, 2015.

[Elo16] Anders Elowsson. Beat tracking with a cepstroid invariant neural network. In *17th International Society for Music Information Retrieval Conference (ISMIR 2016)*, pages 351–.357, 2016.

[Fit10] Derry Fitzgerald. Harmonic/percussive separation using median filtering. In *Proc. of the 13th Int. Conference on Digital Audio Effects (DAFx-10)*, 2010.

[FSJ05] Anders Friberg, Erwin Schoonderwaldt, and Patrik N Juslin. Cuex: An algorithm for extracting expressive tone variables from audio recordings. *Acoustica united with Acta Acoustica*, 93:411–420, 2005.

[HS90] Lars Kai Hansen and Peter Salamon. Neural network ensembles. *IEEE transactions on pattern analysis and machine intelligence*, 12:993–1001, 1990.

[Pol06] Robi Polikar. Ensemble based systems in decision making. *IEEE Circuits and systems magazine*, 6(3):21–45, 2006.

[Pra94] Eric Prame. Measurements of the vibrato rate of ten singers. *The journal of the Acoustical Society of America*, 96(4):1979–1984, 1994.