Automatic characterisation of musical signals: feature extraction and temporal segmentation

S. Rossignol[1,2], X. Rodet[1], J. Soumagne[2], J.-L. Collette[2] and P. Depalle[1]
rossigno@ese-metz.fr,rod@ircam.fr,soumagne@ese-metz.fr
[1]IRCAM 1, place Igor-Stravinsky, 75004 PARIS, FRANCE
[2]Supélec 2, rue Édouard Belin, 57078 METZ, FRANCE

Abstract

This paper presents some results on automatic characterisation of musical and acoustic signals in terms of features attributed to signal segments. These features describe some of the musical and acoustical content of the sound and can be used in applications such as intelligent sound processing, retrieval of music and sound in databases or music editing and labeling. The paper describes a research work at the moment in a very advanced stage but not complete. Applications and results on various examples are presented.

1 Introduction

Three interdependent schemes of segmentation are defined. They correspond to different levels of signal attributes. The data obtained in a given scheme are propagated from lower numbered to higher numbered schemes in order to improve their performance.

Our system is designed to be as general as possible. There is no restriction (see "Moorer (1975)") on the signal to be segmented: the sound can be monophonic music, but it can also be speech or polyphonic music; it can be harmonic or inharmonic (castanets or drums for example); fast notes (less than 100 milliseconds duration for example) and vibrato are taken into account. Data propagation between the different schemes allows our system to be relatively independant of the type of signal. The segmentation and feature extraction system described here is a *work in progress* towards a more general system.

This paper is divided into the six following sections: the segmentation schemes are defined in section 2; techniques used to perform each of them are presented in sections 3, 4 and 5; some dependencies between the schemes are shown in section 6; finally, conclusion and prospects are given in the last section.

2 The three segmentation schemes

• **The first scheme**, named *source* scheme, concerns mainly the distinction between speech, singing voice and instrumental parts such as found on movie sound tracks and on radio broadcasts. Other sounds, such as street sounds and machine noise, will be also considered.
• **The second scheme** refers to features such as silence/sound (energy transients), transitory/steady, voiced/unvoiced (voicing coefficient), harmonic/inharmonic (inharmonicity coefficient), monophonic/polyphonic and vibrato. Most of these characteristics are necessary for *note* and/or *phone* segmentation as done on the third scheme. It is also necessary to suppress vibrato (frequency modulation) from the source signal (singing voice and instrumental part only) to perform the third segmentation scheme (see section 5.2). When speech is identified on the first scheme, there is no need to try to detect vibrato.
• **The third scheme** is segmentation into *note* and/or into *phone* (according to the nature of the sound: instrumental part or singing voice excerpt or speech) or more generally into "stable" parts for monophonic sounds, and into "stable" parts for polyphonic sounds.

This scheme deals also with characterisation of the found segments. If the signal is harmonic and monophonic, the note (as a first step for automatic music transcription) or the phone are also looked for.

Below in the paper it will be noticed that there are dependencies between the results of schemes. Data propagation between schemes are $1 \rightarrow 2 \rightarrow 3$ or $1 \rightarrow 3$ or $2 \rightarrow 3$.
The duration of the segments found by the first scheme can be as long as a few minutes. Commonly, the duration of the segments found by the second scheme is smaller, and the segments found by the third scheme are less than one second long.

3 *Source* segmentation scheme

At the moment, this preliminary work is focusing on the identification of two sound classes. The two classes are: music (that is to say singing voice and/or instrumental part) and speech. The analysis is composed of two steps. Firstly, some features are retained: they measure distinct properties of speech and music. Secondly, these features are combined into several multidimensional classification frameworks. Preliminary results on real data are given.

3.1 Related works

In "Saunders (1996)" a technique is described to discriminate speech from music. Features based on zero-crossing-rate and on energy are used. A trained Multivariate Gaussian classifier is used. 2.4 seconds segment of the data are considered. Experimental results shows performance approching 98 % correct classification.
In "Scheirer and Slaney (1997)" a computer system is described which is capable of distinguishing speech signals from music signals. Thirteen features have been evaluated. Several classifiers have been examined. The best classifier classifies with 1.4 % error when 2.4 second segments of sound are considered.
In "Seck, Bimbot, Zugaj and Delyon (1999)" a technique is described for segmenting a sound track into two classes. The features used are cepstral coefficients and their first order derivatives. A Gaussian Mixture Model is used. On a frame-by-frame basis (Frames are 26 ms long), the error rate is around 20 %.

3.2 First step: feature extraction

Eight features have been examined and retained (see "Scheirer and Slaney (1997)" and "Saunders (1996)"). They are based on the following characteristics:
• **The spectral "flux"**, which is defined as the 2-norm of the difference between the magnitude of the Short Time Fourier Transform (STFT) spectrum evaluated at two successive sound frames. It can be noticed that each evaluation of the STFT is normalized in energy.
• **The spectral centroid** f_c, which is defined as the "balancing point" of the magnitude of the Short Time Fourier Transform $\hat{S}$ spectrum evaluated at a sound frame:

$$f_c = \frac{\sum_{i=0}^{N/2} \hat{S}(i) f_i}{\sum_{i=0}^{N/2} \hat{S}(i)}$$

where f_i is the frequency at the ith bin and where N is the size of the FFT (Fast Fourier Transform).
• **The zero-crossing rate** (ZCR), which is defined as the number of time-domain zero-crossings within a frame.
• **The cepstrum resynthesis residual magnitude** (CRRM), which is defined as the 2-norm of the difference between the magnitude of the Short Time Fourier Transform $\hat{S}$ spectrum evaluated at a sound frame and the "smoothed spectrum" $\hat{M}$ evaluated at the same frame. $\hat{M}$ is obtained using the real cepstrum $\hat{C}$:

$$\hat{C} = \text{real}\left(\text{FFT}^{-1}\left(\log\left(|\hat{S}|\right)\right)\right)$$

$$\hat{C}' = \hat{C} W$$

$$\hat{M} = \exp\left(\text{FFT}\left(\hat{C}'\right)\right)$$

where W is a window with value 1 or 0. Only the n first coefficients (corresponding to smallest positive frequencies) and the n latest coefficients (corresponding to their negative counterpart) of $\hat{C}$ are kept. A 2048-point FFT is performed and $n = 50$ is used for a 44100 Hz sampling rate.
$\hat{M}$ is essentially a low pass filtering of the $\hat{S}$ spectrum, so that a better fit for noise signals than for harmonic signals is obtained.

The eight features are:

- **features 1 & 2:** mean and logarithm of the variance of the spectral "flux"
- **features 3 & 4:** mean and logarithm of the variance of the spectral centroid
- **features 5 & 6:** mean and logarithm of the variance of the ZCR
- **features 7 & 8:** mean and logarithm of the variance of the CRRM

Frames are 18 ms long and are overlapped. Means and variances are computed in a one-second segment. Segments are overlapped.

3.3 Data

The features mentionned above have been computed using two labeled data sets: each set is 20 minutes long, and contains 10 minutes of speech and 10 minutes of music. For each set, 80 % of the labeled samples, selected at random, are used to train the classifiers described in the next section and the remaining 20 % are used to test the performance of the classifiers. The data sets are collected by digitally sampling an FM tuner.

Music can be regarded as a succession of periods of relative stability, notes and phones, in spite of the presence of short signals such as percussions (inducing high-frequency noise). Speech is rather a rapid succession of noise periods, such as unvoiced consonants, and of periods of relative stability, like vowels. Therefore, the selected features give very different values for voiced and unvoiced speech; and they are relatively constant within a segment of musical sound. The variances are higher for speech than for music, whereas the means tends to be greater for music than for speech, as shown in the figure 1.

PLEASE INSERT FIGURE 1 HERE (nuages4eps.eps)

3.4 Second step: classification

In order to identify the segments, four classification methods have been tested. Only three have been retained:

- Central Clustering (CC) which has given not conclusive results.
- a Gaussian Mixture Model (GMM) classifier. The mixture parameters (that is to say the mean vector y and the covariance matrix Σ) of each class are computed using the N training vectors x, according to the formulas:

  $$y = \frac{1}{N}\sum_{i=1}^{N} x_i \qquad\qquad \Sigma = \frac{1}{N}\sum_{i=1}^{N} (x_i - y)(x_i - y)^T$$

  where T denotes transposition. The length of vectors y and x_i is the number n of features used and Σ is a $n \times n$ matrix.

  Then, a given point x_j to be classified belongs to the class α if:

$$\frac{\pi_\alpha}{(2\pi)^{\frac{n}{2}}\sqrt{d\Sigma_\alpha}}\exp\left(-\frac{1}{2}(x_j-y_\alpha)^T\Sigma_\alpha^{-1}(x_j-y_\alpha)\right) > \frac{\pi_\beta}{(2\pi)^{\frac{n}{2}}\sqrt{d\Sigma_\beta}}\exp\left(-\frac{1}{2}(x_j-y_\beta)^T\Sigma_\beta^{-1}(x_j-y_\beta)\right)$$

where d denote the determinant and $^{-1}$ the inverse matrix. Each set is composed of as many music vectors as speech vectors, so that the *a priori* probabilities π_α and π_β are equal and can be eliminated in the previous equation.

- the k-Nearest-Neighbours (kNN) classifier, with $k = 7$. For a given point to classify, its k nearest neighbour training vectors are determined. If most of these training vectors belong to class α, the given point is classified in the class α.
- Neural Networks (NN) classifier (see "SNNS[1] (1995)"). We use a fully connected Neural Network composed of three layers. For classification with only features one and two, the hidden layer contains six units. When six features are used, the hidden layer contains eight units.

3.5 Performance

The percentage of missclassified segments is given in tables 1, 2 and 3. Table 1 shows results when using only the features 1 & 2. Table 2 presents the results for each couple of features when using the kNN classifier ($k = 7$). Table 3 shows the best results obtained, when using the six first features. Column 1 lists training errors obtained by comparing the results of a manual segmentation and the results of the training step. Next, testing errors, in column 2, are errors done when using training parameters obtained for the set number i to segment the test part of the same set. For the cross-testing, on column 3, the training part of the set number i is used, and the test is operated on the testing part of the set number j, with $j \neq i$. Only the average percentage values are presented.

PLEASE INSERT TABLE 1 HERE

PLEASE INSERT TABLE 2 HERE

PLEASE INSERT TABLE 3 HERE

The tables indicate that the retained features are useful to separate the two classes. In "Scheirer and Slaney (1997)" only classical classifiers (GMM and kNN) have been used. Results obtained here with those classifiers are similar to those obtained in "Scheirer and Slaney (1997)": 5.9 % error. There is noticeable improvement when using Neural Network classifiers since the error rate drops to 4.7 % (see table 3). When cross-testing error is evaluated in 2.4 second segments, the error rate, when using Neural Networks classifiers, drops to 1.3 %. But the rate of cross-testing errors indicates that generalization is not today satisfactory: the data sets are too small, especially for music (various styles are represented, but not all the available styles!). Tests on larger data sets are presently done in order to provide better insight.
Some improvements are envisioned: the use of the union of several gaussians to model each class (especially the music class), the use of more performant neural netwoks classifiers (more hidden layers)...

However, this work, which is a first step for the classification of sounds into three or more classes, provides very promising results.

4 *Feature* scheme

4.1 Features in use

- presence of vibrato/absence of vibrato (see the section 4.2): it is based on the fundamental frequency f_0 trajectory. The f_0 trajectory is obtained using an IRCAM software, named *f0*[2]: for more details see "Serra (1989)", "Doval (1994)".

[1] HTTP://WWW.INFORMATIK.UNI-STUTTGART.DE/IPVR/BV/PROJEKTE/SNNS/SNNS.HTML
[2] HTTP://WWW.IRCAM.FR/EQUIPES/ANALYSE-SYNTHESE/ADDITIVE/INDEX-E.HTML

- silence/sound: it is based on the energy trajectory.
- harmonic/inharmonic: it is based on the measure of the inharmonicity of sinusoidal partials. An IRCAM software, named *additive*[2], computes the trajectories of all the N partials between f_0 and half sampling frequency $f_e/2$. Then, the relative frequency deviation between the theoretical location of the nth partial and its practical location is defined in this work as: $h_n = |f_n - n.f_0|/(n.f_0)$, with $n \epsilon [2, N]$. The final measure of the inharmonicity is:

$$H(i) = \sum_{n=2}^{N} h_n$$

- voiced/unvoiced: it is based on a global voicing coefficient. It uses the magnitude of the Short Time Fourier Transform spectrum $\hat{S}$ computed on a 20 ms windowed signal frame, and the value of f_0 for this frame. A voicing coefficient is defined for each of the N partials between f_0 and $f_e/2$. For each partial n, a $2\Delta f$ bandwith is retained. The "signal magnitude spectrum" $|\hat{S}|$ is truncated between $nf_0 - \Delta f$ and $nf_0 + \Delta f$ and normalized in energy. Similarly, the "frame window's magnitude spectrum" $|\hat{W}|$ (magnitude of the Short Time Fourier Transform spectrum of the temporal window) is truncated between $-\Delta f$ and Δf and normalized in energy. The $2\Delta f$ bandwith is chosen for keeping the largest part of the energy of the mainlobe of $|\hat{W}|$. A high value of v_n, the correlation coefficient of these two truncated and normalized spectra, confirms the presence of a sine wave (corresponding to a partial). Then, the proposed global voicing coefficient is:

$$V = \sum_{n=1}^{N} v_n$$

- monophonic/polyphonic. The softwares $f0$ and *additive* assume that the signal is harmonic, that is to say monophonic. An IRCAM software, named *HMM* (see "Depalle and all (1993)"), track partials by identifying functions of time as underlying trajectories in successive set of spectral peaks. This is done by a purely combinatorial Hidden Markov Model. A partial trajectory is considered as a time-sequence of peaks which satisfies continuity constraints on parameter slopes. This method allows frequency line crossing. It allows also the detection of several fundamental frequencies by grouping partials into several harmonic patterns. Thus it can allow the detection of the polyphonic character of the sound.

4.2 *Vibrato* detection and suppression algorithm

An algorithm to detect and to suppress the vibrato is proposed below. To model the vibrato (that is to say to find its frequency and its amplitude) is useful for musical processing of the sound, and also to characterize the sound. Suppression of the vibrato from the f_0 trajectory is helpful for the procedure of segmentation into *note* and/or *phone* (see section 5.2).
The principle of the vibrato detection and suppression algorithm is illustrated in figure 2. Firstly, the f_0 trajectory local maxima are detected (max list is formed). The $maxinterp$ trajectory is obtained by interpolating maxima points. All the temporal distances between two successive local maxima are calculated ($distmax$ list). Secondly, the variance of $distmax$ ($Vmax$) is calculated. The number of periods $distmax$ comprised between $0.15\,s$ and $0.25\,s$ (these values correspond to a $4\,Hz$ to $6.7\,Hz$ vibrato) is evaluated as a percentage $Pmax$. The same processings are done for the local minima. When such a vibrato exists, most of the vibrato periods are belonging to the small range between $0.15\,s$ and $0.25\,s$, then variances have low values and percentages get high values. Thirdly, if a vibrato is detected, the frequency distances between $maxinterp$ trajectory and $mininterp$ trajectory at all the instants are computed ($distfreq$ list). If their relative mean: $Mfreq = \text{mean}\left(\frac{maxinterp - mininterp}{(maxinterp + mininterp)/2}\right)$ is great, the vibrato is significant and have to be suppressed from the f_0 trajectory.
In fact, in presence of significant vibrato, the *note* and/or *phone* scheme features based on f_0 trajectory (features 1, 2, 8, 9: see section 5.2) fail. The new f_0 trajectory, without vibrato, is calculated as the

average of the *maxinterp* and *mininterp* trajectories. Then, the *note* and/or *phone* scheme features based on this new f_0 trajectory are much more successful.

PLEASE INSERT FIGURE 2 HERE (detectvibra3eps.eps)

4.3 *Vibrato* detection and suppression algorithm: results

Results are given in the table 4. They are obtained for a flute excerpt (Saint Saëns, Le Cygne), without vibrato; and for a singing voice excerpt, with vibrato.

PLEASE INSERT TABLE 4 HERE

We give on figure 3 the original f_0 trajectory (left side) for a part of the singing voice excerpt and the corresponding new f_0 trajectory (right side). The notes have been reported and the segmentation marks have been manually put. It can be seen that the amplitude of the vibrato is sometimes equal to the gap between two successive notes.

PLEASE INSERT FIGURE 3 HERE (vibraeps1.eps)

5 *Note* and/or *phone* segmentation scheme

Until now only the segmentation of monophonic sounds have been studied in this project. The analysis is composed of four steps. The first step is to extract a large set of features. A feature will be all the more appropriate as its time evolution presents strong and short peaks when transitions occur, and as its variance and its mean values remain at very low levels in steady part. Secondly, each of these features is automatically thresholded. Thirdly, a final decision function based on the set of the thresholded features is built and provides the segmentation marks. Lastly, each segment is characterised in terms of "the played note", "harmonic or noise segment" and so on.

5.1 Related works

A short survey on segmentation of musical sounds works and research is given here. Here is a non exhaustive list of the various techniques used.
In "Moorer (1975)", the feature used to segment the sounds into notes is the energy derivative.
In "Basseville and Nikiforov (1993)", several techniques are described to segment signals. In "Lepain and André-Obrecht (1995)", these techniques are adapted for speech and musical signals.
In "Smith (1994)", features based on onset/offset filters are presented. The filters used are based on the ideas applied to visual processing in "Marr and Hildreth (1980)".
In "Masri and Bateman (1996)", a feature is described which is based on the energy and High Frequency Content (or centroid) of the spectrum.
In "Hadja (1996)", an algorithm based on the energy and centroid of the spectrum is presented. It allows segmentation of sounds and labeling of each segment with the following labels : attack, transition, steady state or decay.
In "Solbach and Wöhrmann (1996)" a new feature is presented, based on a wavelet approach.
In "Jehan (1997)", the Brandt algorithm (Generalized Likelihood Ratio (GLR) test of Brandt) is applied to musical signals.
In "Dubnov and Tishby (1997)", features based on higher order moments are presented.

5.2 First step: feature extraction

For the *note* and/or *phone* scheme, the features in use are:

- **features 1 & 2:** derivative and relative derivative of f_0: At the instant i, the derivative of the function f is defined as $d(i) = |f(i+1) - f(i-1)|$ and its relative derivative is defined as $\delta(i) = d(i)/f(i)$.

  PLEASE INSERT FIGURE 4 HERE (vibraeps2.eps)

  In the figure 4 it can be seen why, in case of significant vibrato, features based on f_0 trajectory fail and why it is necessary to suppress vibrato from this trajectory. It can be seen that before vibrato suppression the peaks due to transitions are merged with peaks caused by vibrato, and that after vibrato suppression the feature presents strong peaks when transitions occur. On the right side, a threshold has been manually put (horizontal line). The natural transitions marks (dashed vertical lines) have been manually put on the left side of the figure 3 and have been moved on the right side of the figure 4. It appears that each mark coincides with a local maximum of the derivative of the new f_0 trajectory when this maximum is greater than the indicated threshold. Therefore transitions can be detected by thresholding the derivative of f_0 after vibrato suppression.

- **features 3 & 4:** derivative and relative derivative of energy.
- **feature 5:** derivative of the inharmonicity coefficient $H(i)$ (see section 4.1).
- **feature 6:** derivative of the voicing coefficient $V(i)$ (see section 4.1).
- **feature 7:** spectral "flux": Already defined (see the *source* segmentation scheme).
- **feature 8:** "distance between two statistical models" method, used on the f_0 trajectory (see "Cerveau (94)"). This feature is able to detect pitch transition between two notes. Two segments of this trajectory (usually of duration 50 ms) are considered: one before and one after the current analysis time i (in samples). We assume that each of these two observation is obeying a stochastic process. The probability $p(i)$ that the distance between the two models exceed a threshold S is computed. Commonly, we consider that the models are obeying normal processes. Accordingly, the probability $p(i)$ is:

$$p(i) = 1 - \frac{1}{\sqrt{2\pi}\sqrt{\sigma_1^2 + \sigma_2^2}} \int_{-S}^{+S} \exp\left[-\frac{1}{2}\left(\frac{y-(m_2-m_1)}{\sqrt{\sigma_1^2+\sigma_2^2}}\right)^2\right] dy$$

  where m_1 and σ_1^2 are the mean and the variance of the first observation, and where m_2 and σ_2^2 are the mean and the variance of the second observation. It can be noticed that S is related to a musical characteristic, i.e. a pitch gap (such as a quarter-tone).

- **feature 9:** "detection of abrupt changes using AR modelling" method, used on the f_0 trajectory. Two segments are considered. One covers fifty milliseconds preceding the current sample j. The other one extends from the previously detected change at sample i to current sample j ($i < j$). An Auto-Regressive (AR) model is calculated on each observation, and their *Mutual Entropy* is estimated. The *Cumulative Entropy* (CE) is computed. When the difference between the maximum CE on the i to j interval and the CE at j is greater than a threshold λ, a change is detected and the current change point is moved to j. This method is known as the divergence algorithm: for more details, see "Basseville and Nikiforov (1993)" and "Jehan (1997)", "Lepain and André-Obrecht (1995)". It can be noticed that usually this method is not used on a parameter trajectory but directly on the sound signal.

The energy (features 3 and 4) and spectra (features 6 and 7) are calculated on overlapped frames. The chosen frame duration is typically 20 ms. The aforementioned features are sampled at 100 Hz. The "distance between two statistical models" and the "detection of abrupt changes using AR modelling" methods can also be used on energy trajectory.
It can be noticed that some of these features (energy derivatives, spectral flux) can be used directly to segment polyphonic sounds. The use of the others features is extensible to polyphonic music if one assumes that the step of multiple f_0 tracking (sources separation) has been conducted successfully.

5.3 Second step: thresholding

Only feature 9 is based on a decision algorithm. For the others, discrimination between meaningful peaks and noise peaks must be done. To discriminate between these two classes, a threshold is automatically estimated. For our system to be as automatic as possible, thresholds should not be empirically chosen as it is commonly done (see "Moorer (1975)", "Masri and Bateman (1996)").
Several automatic thresholding techniques have been implemented and tested (see "Sahoo and all (1988)"). These techniques are popular in digital image processing. In our case, peaks due to transitions are rare and their variance is great. The performance of the thresholding techniques is severely degraded when the *a priori* probability of one class is very small and when the variance of the same class is much greater than the variance of the other class, as in our case. Therefore more performant methods have to be used, for example based on Neural Networks.
At the moment, we have retained the 3σ rule. Assuming a normal distribution for the noise, one can determine σ when retaining for example $n = 90\%$ of the smallest samples of the feature. The method is relatively robust according to n. Then, the value of the threshold is 3σ (the peaks greater than 3σ are considered as transition peaks). Thresholding gives a function with values 0 (steady state) or 1 (a transition is detected).

5.4 Third step: final decision function

A decision function is computed, based on the particular decisions obtained for each feature. The decision functions (sequences of 0 and 1) are added (giving value greater than one when several of them indicate transitions exactly at the same sampled time). The resulting signal S corresponds to a kind of variable density pulse signal. Actually, marks are scattered because transitions are not instantaneous. In addition, we use windows: this leads to temporal resolution limit; and the signal may have reverberation, or the instrument may be not really monophonic: in some cases the end of a note is mixed with the beginning of the next note. Some of the features react rather at the beginning of a transition, others react rather at the end of a transition, and the other ones react several times during a transition. Therefore, the signal S has to be modified as follows. When the distance between two segmentation marks (pulses) is smaller than T ms ($T = 50$ ms), these segmentation marks are replaced by a unique mark, the weight v of which is the sum of the two previous values v_1 and v_2, and the position p of which is the corresponding centre of mass:

$$v = v_1 + v_2 \qquad p = \frac{p_1 v_1 + p_2 v_2}{v_1 + v_2}.$$

The original pulse signal S is modified step by step following this way. The result is a weighted pulse signal where weight corresponds to the confidence granted to the presence of a transition. The optimized localisation of these pulses (due to the centre of mass calculation) gives good approximation of the exact time of transition. Then, a threshold is applied to eliminate the smallest segmentation marks (false alarms).

This final decision depends on the current application. As an example, it is possible to sing several consecutive vowels with the same pitch: segmentation according to pitch must not lead to different segments and in the same case segmentation into phones must separate exactly the vowels. One can also imagine several pitches sung with the same vowel. Then, for the final step, some features among the previous list are selected: the most appropriate for the current application (segmentation into notes or phones or other combination) are chosen. Related to the previous example (singing voice), features 1, 2, 3, 4, 5, 8 and 9 correspond to pitch segmentation, and features 6 and 7 correspond to phone segmentation. We use several features to improve the quality of the *note* and/or *phone* segmentation, that is to say reducing the number of false alarms (it is assumed that during a steady state few features react exactly at the same time, that is to say giving a false alarm at this instant), and to avoid missing marks (we assume that at a given transition few features do not react, that is to say giving a missing mark at this instant). This kind of "data fusion" should improve the robustness of the system since each individual feature may not provide a very robust decision.

Others features based on methods such as higher order statistics and cepstral coefficients are under study. Some of them can be directly added to our system: see "Masri and Bateman (1996)", "Smith (1994)", "Basseville and Nikiforov (1993)" (Generalized Likelihood Ratio (GLR) test of Brandt), "Dubnov and Tishby (1997)".

5.5 Characterisation of each segment

Finally, each segment has to be characterized in terms of note, harmonicity, amount of noise component, and so on.

5.6 A graphical interface

A graphical interface have been developped on Unix (SGI®, alpha®, Sun®) and on Microsoft® Visual C++ using the Xforms library (see "Zhao and Overmars (1996)"[3]). The different steps of the analysis are displayed, and some parameters (such as frame's length) can be ajusted.
Results for the previous Saint Saëns flute excerpt are given on figures 5, 6 and 7.
On figure 5, the features are shown (first step); on figure 6, the thresholded features are presented (second step); and on figure 7, the result of the final decision is given (third step).

PLEASE INSERT FIGURE 5 HERE (es_fig5_3.eps)

PLEASE INSERT FIGURE 6 HERE (es_fig6_2.eps)

PLEASE INSERT FIGURE 7 HERE (es_fig7_1.eps)

5.7 Results

For several music excerpt, the result when using the whole system is shown in the table 5. And the result when using only one feature and when not using the vibrato detection and suppression algorithm is shown on the table 6. Column two lists the number of transitions to detect. Column three lists the number of good detections (gd). And column four lists the number of false alarms (fa). It can be seen that using more features improves the robustness of the system.
The violin excerpt is very noisy since it has been recorded at a very low level. In addition, one can hear the noise of the score pages as they are turned. The result for the song excerpt with significant vibrato, is shown on the last row. It can be seen that the vibrato detection and suppression algorithm provides a good performance.

PLEASE INSERT TABLE 5 HERE

PLEASE INSERT TABLE 6 HERE

6 Dependencies between the three segmentation schemes

PLEASE INSERT FIGURE 8 HERE (depend.eps)

Some dependencies between the three schemes are shown on the figure 8.

• For the vibrato, the algorithm is working as follows:
When speech is identified by the first scheme, there is no need to try to detect vibrato. Then the variable VIBRATO is set to OFF. The "suppression of vibrato" tool is not used when features are extracted for the third scheme.
When music is identified, the "detection of vibrato" tool is used on the second scheme. If vibrato is detected the variable VIBRATO is set to ON. In that case, vibrato is suppressed from f_0 trajectory and the four features based on f_0 are estimated using the new f_0 trajectory.

[3] HTTP://BRAGG.PHYS.UWM.EDU/XFORMS

Whenever a vibrato is present, the mainlobe of a partial is all the more distorted as its order is greater. Therefore, the voicing coefficient is evaluated using only the first partials (see section 4.1), and the spectral flux is evaluated using a bandwith narrower than $[0\ f_e/2]$.

• For the inharmonicity coefficient the algorithm is working as follows:
When analysing inharmonic sounds, such as castanets, the feature based on inharmonicity coefficient does not present strong and short peaks at transitions. Therefore, this feature can not be used to provide the *note* and/or *phone* segmentation marks. Since inharmonicity can not be used to detect transitions when the signal is inharmonic the variable USEINHARMONICITY (figure 8) is set to OFF.

7 Conclusion and prospects

"Segmentation" and "feature extraction" are aimed at being a first step for sound signal representation, coding, transformation, indexation, multimedia, and also for source separation. The system used to separate speech and music provides good and robust results. The performance of the *note* and/or *phone* segmentation system, in case of monophonic sounds, is considered as good, and some problems caused by complicated signals (presence of vibrato, percussions) have been solved in large part.

All the computational links between the three segmentation schemes described above have to be done. For the first scheme, others features are under implementation or study. We are searching for features which measure distinct properties of instrumental sound, singing voice and others sounds. A larger database is needed. For the second scheme, the detection silence/signal and the detection monophonic/polyphonic have still to be done. For the third scheme, others features are envisioned, for example based on cepstral coefficients, or higher order moments. Others are under implementation ("Masri and Bateman (1996)", "Smith (1994)"). The correlation between the features is being studied, in order to reduce redundancies. We have to improve the robustness of the two last steps of the analysis, for example by using Neural Networks in order to take better decisions. Polyphonic sound temporal segmentation remains to be explored.

Acknowledgements

This work is supported by CNET/RENNES (Centre National d'Études des Télécommunications), FRANCE.

References

[BN93] Michèle Basseville and Igor V. Nikiforov, *Detection of abrupt changes*, PTR Prentice-Hall, 1993.

[Bra83] A. Von Brandt, *Detecting and estimating parameters jumps using ladder algorithms and likelihood ratio test*, IEEE Transactions on Acoustics, Speech, and Signal Processing (ICASSP'83), 1983, pp. 1017 – 1020.

[Cer94] Laurent Cerveau, *Segmentation de phrases musicales à partir de la fréquence fondamentale*, Stage de DEA, IRCAM, Juillet 1994.

[DGR93] Philippe Depalle, G. Garcia, and Xavier Rodet, *Tracking of partials for additive sound synthesis using hidden markov models*, IEEE Transactions on Acoustics, Speech, and Signal Processing (ICASSP'93), 1993.

[Dov94] Boris Doval, *Estimation de la fréquence fondamentale des signaux sonores*, Ph.D. thesis, Université de Paris VI, Mars 1994.

[DT97] Shlomo Dubnov and Naftali Tishby, *Analysis of sound textures in musical and machine sounds by means of higherorder statistical features*, IEEE Transactions on Acoustics, Speech, and Signal Processing (ICASSP'97), 1997.

[Haj96] John Hajda, *A new model for segmenting the envelope of musical signals: The relative salience of steady state versus attack, revisited*, Audio Engineering Society, november 1996.

[Jeh97] Tristan Jehan, *Musical signal parameter estimation*, Tech. report, IFSIC – CNMAT, 1997.

[LAO95] Philippe Lepain and Régine André-Obrecht, *Micro-segmentation d'enregistrements musicaux*, Journées d'Informatique Musicale (JIM), 1995, pp. 81 – 90.

[MB96] Paul Masri and Andrew Bateman, *Improved modelling of attack transients in music analysis/resynthesis*, International Computer Music Conference (ICMC'96), 1996.

[MH80] D. Marr and E. Hildreth, *Theory of edge detection*, Proceedings of the Royal Society of London, 1980.

[Moo75] James A. Moorer, *On the segmentation and analysis of continuous musical sound*, Ph.D. thesis, Stanford University, 1975.

[Rap95] SNNS - S*tuttgart* N*eural* N*etwork* S*imulator - user manual*, Tech. report, University of Stuttgart, 1995.

[RRS+98] S. Rossignol, X. Rodet, J. Soumagne, J.-L. Collette, and P. Depalle, *Feature extraction and temporal segmentation of acoustic signals*, 1998.

[Sau96] John Saunders, *Real-time discrimination of broadcast speech/music*, IEEE Transactions on Acoustics, Speech, and Signal Processing (ICASSP'96), 1996, pp. 993 – 995.

[SBZD99] Mouhamadou Seck, Frédéric Bimbot, Didier Zugaj, and Bernard Delyon, *Two-class signal segmentation for speech/music detection in audio tracks*, Eurospeech, september 1999.

[Ser89] Xavier Serra, *A system for sound analysis/transformation/synthesis based on a deterministic plus stochastic decomposition*, Ph.D. thesis, Stanford University, October 1989.

[Smi94] Leslie S. Smith, *Sound segmentation using onsets and offsets*, Journal of New Musical Research, vol. 23, Numéro 1, March 1994, pp. 11 – 23.

[SS97] Eric Scheirer and Malcolm Slaney, *Construction and evaluation of a robust multifeatures speech/music discriminator*, IEEE Transactions on Acoustics, Speech, and Signal Processing (ICASSP'97), 1997, pp. 1331 – 1334.

[SSW88] P. K. Sahoo, S. Soltani, and K. C. Wong, *A survey of thresholding techniques*, Computer Vision, Graphics, and Image Processing, vol. 41, 1988, pp. 233 – 260.

[SW96] Ludger Solbach and Rolf Wöhrmann, *Sound onset localixzation and partial tracking in gaussian white noise*, International Computer Music Conference (ICMC'96), 1996.

[ZO96] T. C Zhao and Mark Overmars, *Forms library - a graphical user interface toolkit for* X, Tech. report, University of Wisconsin-Milwaukee - Utrecht University, july 1996.

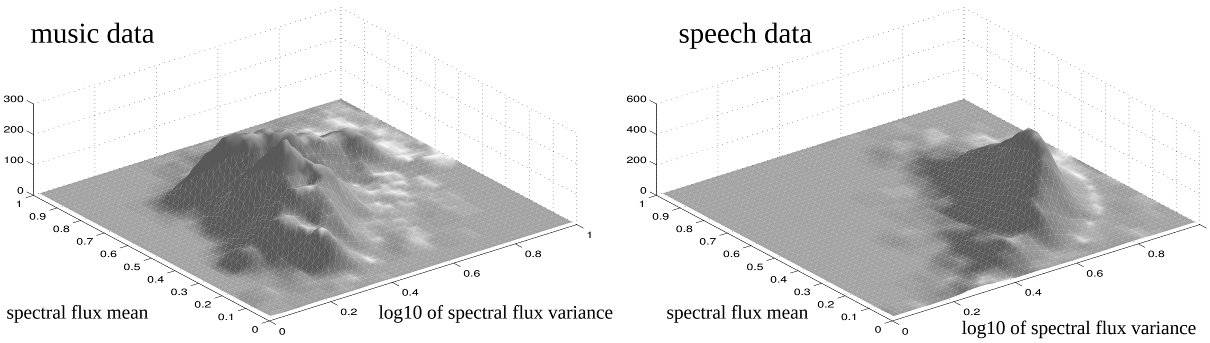

Figure 1: Normalized features 3D histogram (first set).

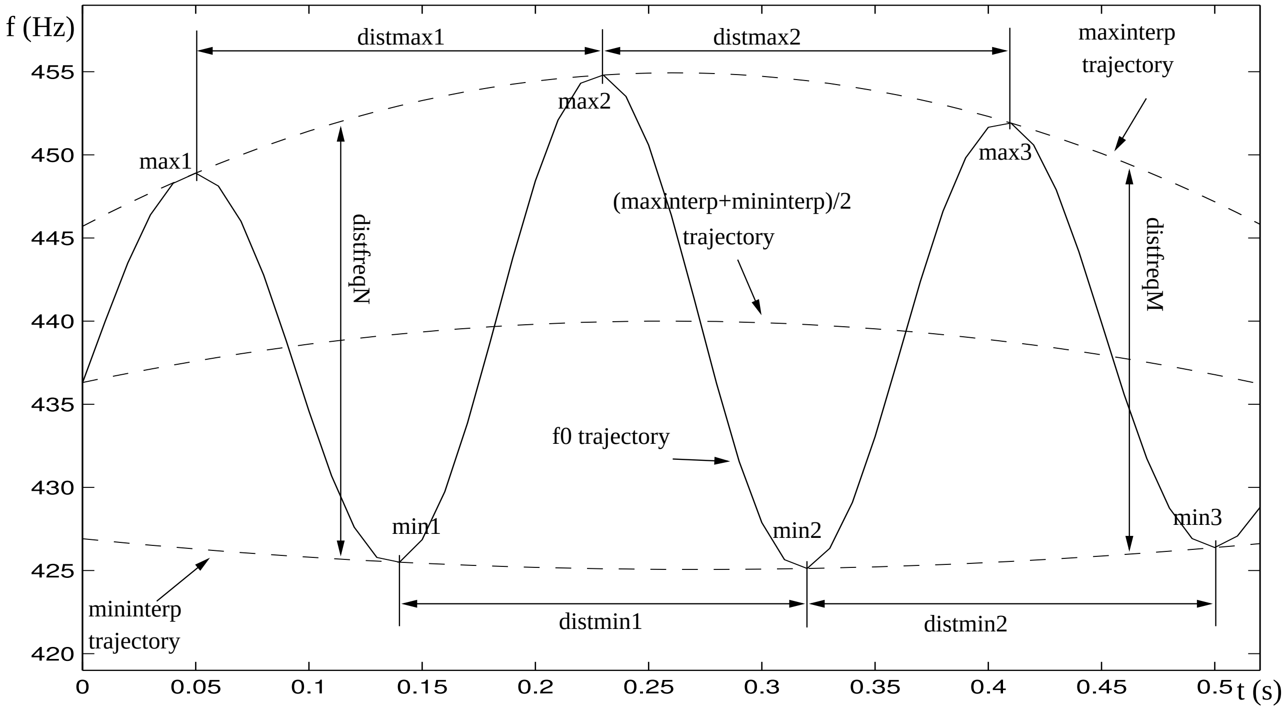

Figure 2: Vibrato detection and suppression algorithm

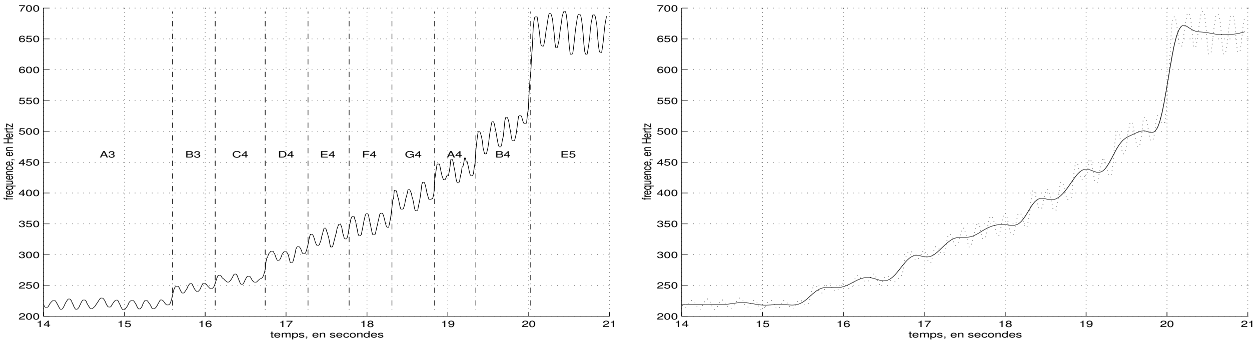

Figure 3: f_0 trajectory before and after vibrato suppression

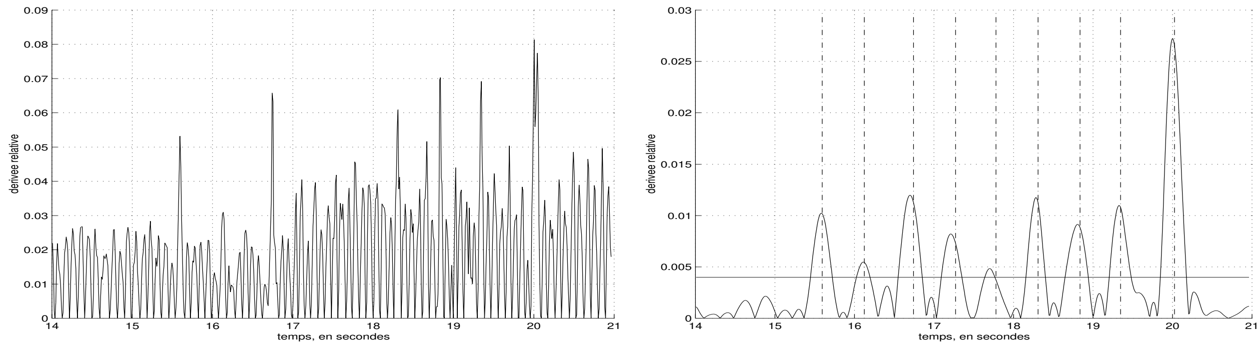

Figure 4: derivative of f_0 trajectory before and after vibrato suppression

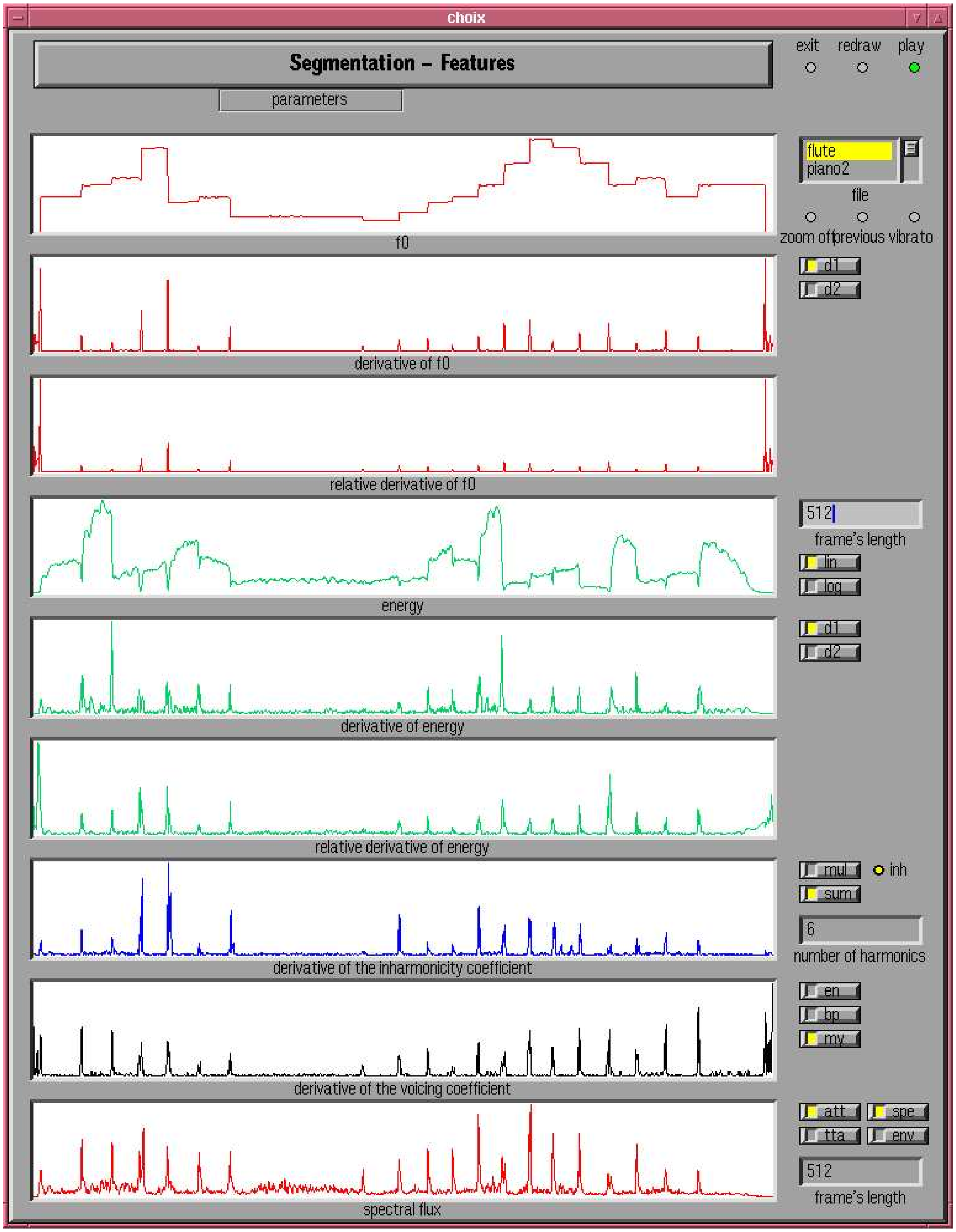

Figure 5: First step: features calculation. On the canvases are shown, from top to bottom, f_0 trajectory, derivative of f_0 trajectory, relative derivative of f_0 trajectory, energy trajectory, derivative of energy trajectory, relative derivative of energy trajectory, derivative of the inharmonicity coefficient, derivative of the voicing coefficient and spectral flux. We can modify the frame length for energy, voicing coefficient and spectral flux calculation. Other parameters can be accessed with the "parameters" menu.

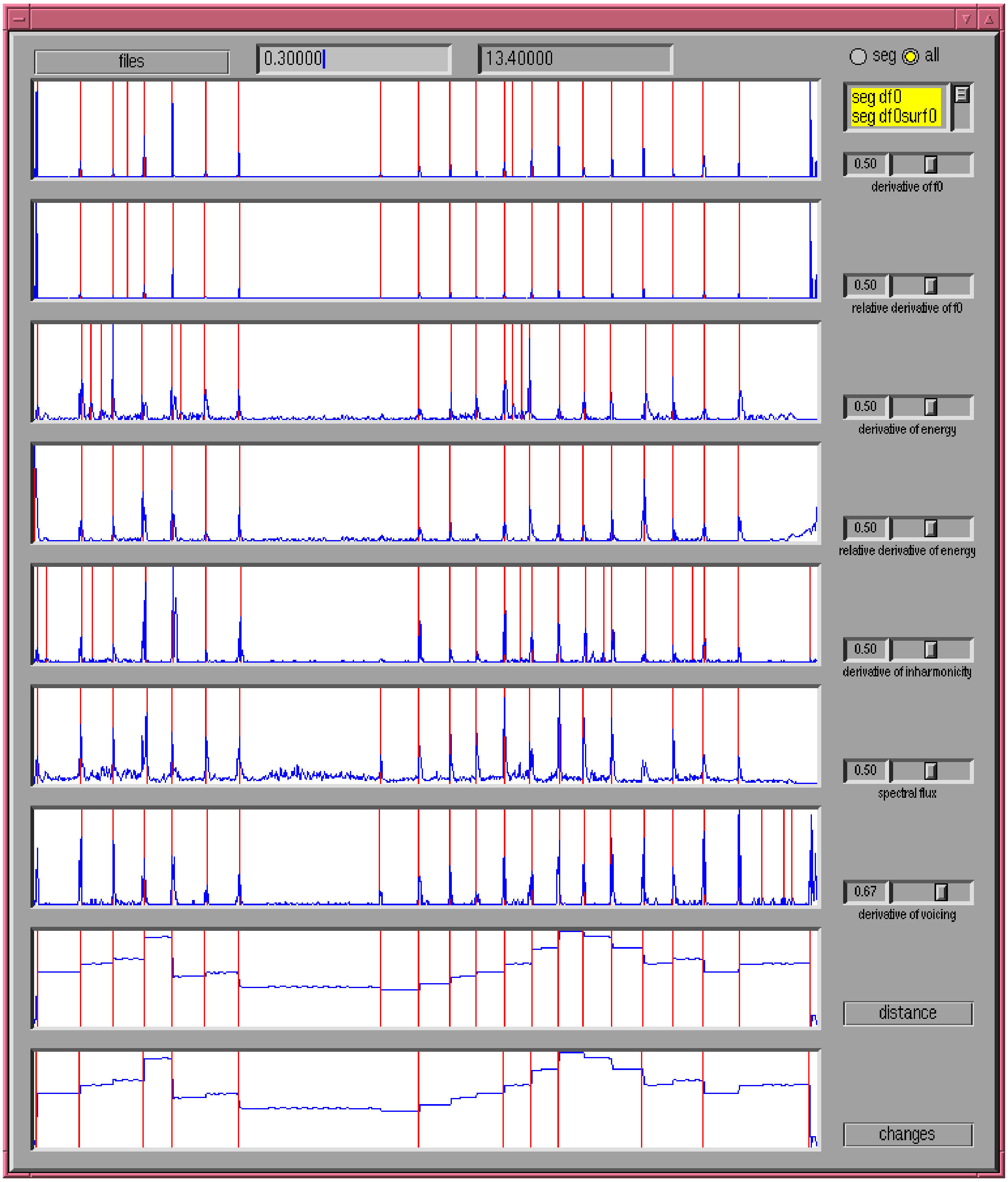

Figure 6: Second step: thresholding. Automatically thresholded features are displayed. The dark trajectory is the feature, and the lighter vertical lines are the found segmentation marks. On the canvases are shown, from top to bottom, derivative of f_0 trajectory, relative derivative of f_0 trajectory, derivative of energy trajectory, relative derivative of energy trajectory, derivative of the inharmonicity coefficient, derivative of the voicing coefficient, spectral flux, result of the "distance between two statistical models" method applied on f_0 trajectory and result of the "detection of abrupt changes using AR modellisation" method applied on f_0 trajectory. For each feature false alarms and missing marks are observed. It can be seen that as expected the features do not react in the same manner.

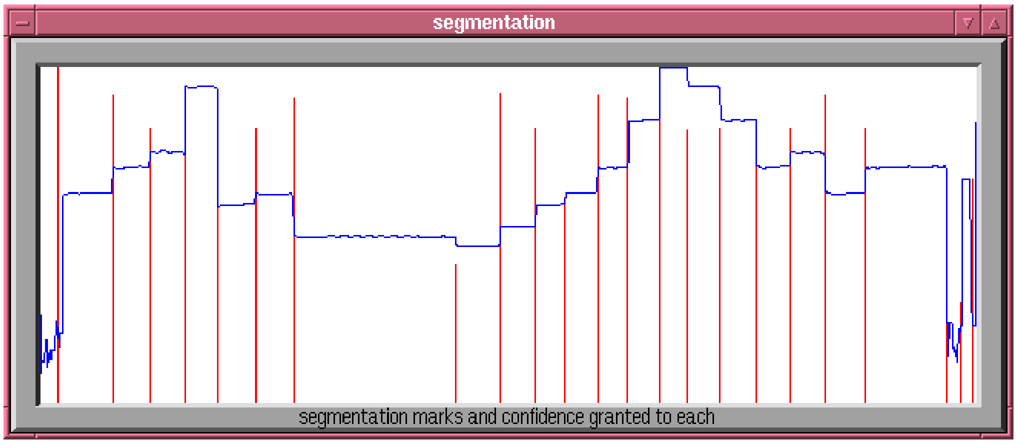

Figure 7: Result of the third step: concatenation of the previous results. The dark trajectory is the f_0 trajectory, and the vertical lighter lines are the final segmentation marks. The height of vertical lighter lines represent the confidence granted to each segmentation mark.

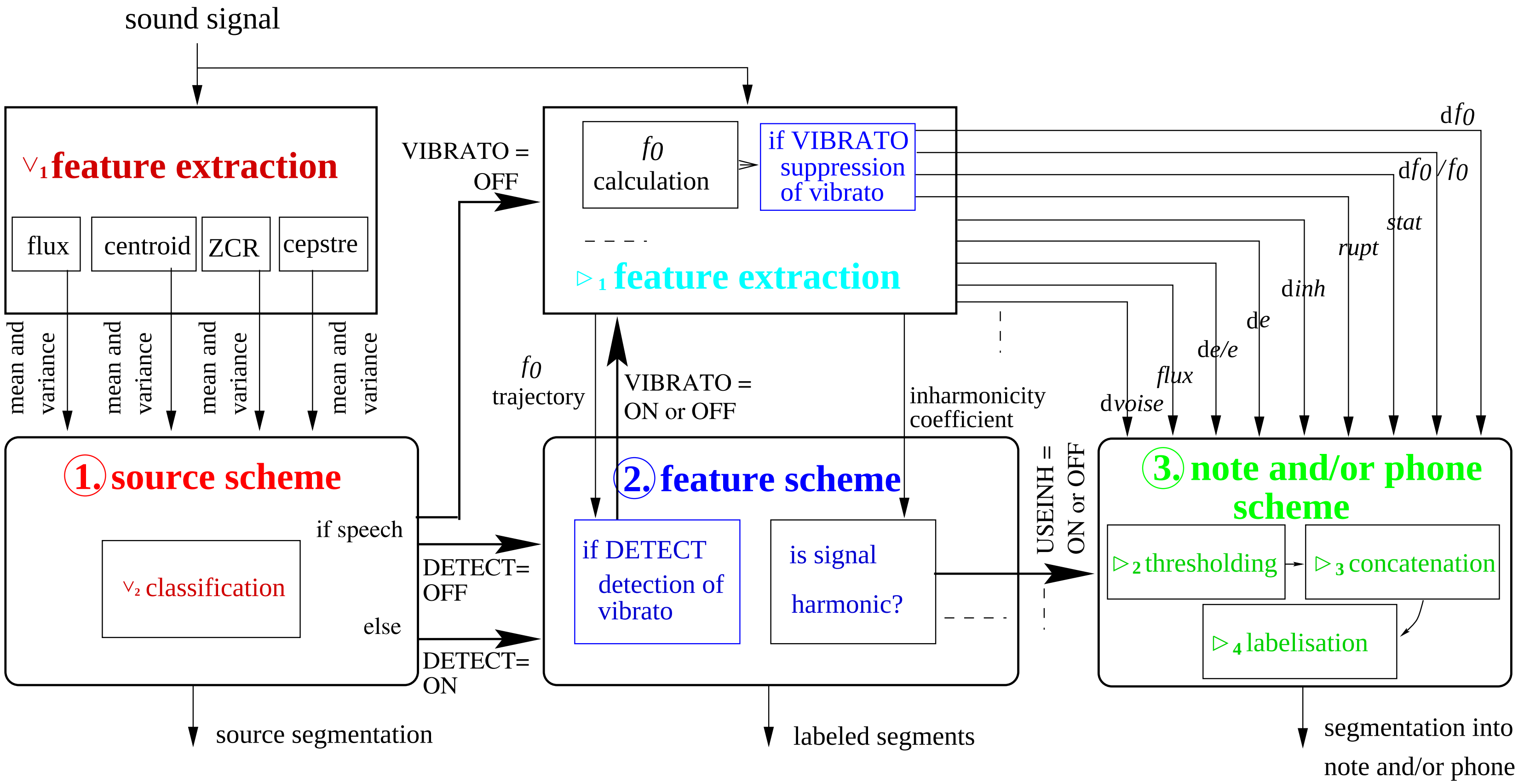

Figure 8: Some dependencies between the schemes

	training	testing	cross-testing
GMM	8.6 %	8.8 %	11.3 %
kNN (k=7)	4.7 %	5.4 %	9.1 %
NN	5.5 %	5.2 %	7.4 %

Table 1: Percentage of missclassified segments when using features 1 & 2

	training	testing	cross-testing
flux mean and log(variance)	4.7 %	5.4 %	9.1 %
centroide mean and log(variance)	7.3 %	8.7 %	12.7 %
ZCR mean and log(variance)	7.4 %	8.3 %	13.2 %
CRRM mean and log(variance)	8.2 %	10.6 %	13.8 %

Table 2: Percentage of missclassified segments for each couple of features when using the kNN (k=7) classifier

	training	testing	cross-testing
GMM	4.0 %	3.6 %	9.3 %
kNN (k=7)	1.3 %	1.5 %	5.9 %
NN	1.9 %	1.7 %	4.7 %

Table 3: Percentage of missclassified segments when using the six first features

	Vmax	Vmin	Pmax	Pmin	Mfreq
song (vibrato)	0.0010	0.0012	85 %	84 %	0.087
flute (no vibrato)	0.0090	0.0100	50 %	46 %	0.032

Table 4: Vibrato detection results

	nber of transitions	nber of gd	nber of fa	
flute	21	21	0	86 gd
clarinet	30	30	2	98 % of gd
violin	16	15	2	and 4 fa
song	21	20	0	

Table 5: Using the whole system ("gd" stand for good detections and "fa" stand for false alarms)

	nber of transitions	nber of gd	nber of fa	
flute	21	21	3	87 gd
clarinet	30	29	2	99 % of gd
violin	16	16	15	and 29 fa
song	21	21	9	

Table 6: Using only one feature: derivative of f_0 ("gd" stand for good detections and "fa" stand for false alarms)