2002-2003

INTERNSHIP REPORT

Leonardo-Internship
IRCAM, 1st December 2002 to 30th April 2003

FUNDAMENTAL FREQUENCY ESTIMATION

Matthias Krauledat

Head of the department "Analyse-Synthèse" (IRCAM)	:	Xavier	Rodet
Tutor (IRCAM)	:	Axel	Röbel

Contents

Chapter 1

About the internship

My internship took place at the IRCAM (Institut de Recherche et de Coordination Acoustique/Musique), (Institute of research and coordination of acoustics and music), where I worked in the research team Analyse-Synthese under the direction of Xavier Rodet. My instructor was Axel Röbel.

1.1 Organizational structure of the IRCAM

Since its foundation in 1971 by Pierre Boulez, on request of the french president Georges Pompidou, the IRCAM has the status of a private and independent organization (association relevant du droit privé (loi 1901)). Its council of administration consists of seven members of representatives of public institutions, namely of the ministries of culture and research, of the Centre Pompidou and the Centre National de la Recherche Scientifique (CNRS). Chairman of this council is the chief executive of the Centre Pompidou; Bernhard Stiegler is the chief executive of the IRCAM since January 2002.

The IRCAM was founded with the objective to provide various fields of interaction between scientifical research, technology development and the musical creation process. Still, the main aims are to support the creation of contemporary music by developing software and know-how for composers as well as to advance in acoustical research areas. The idea to combine these two areas (research and musical creation) has lead to the development of models and utilities in different aspects of sound processing (language, man-machine-interfaces, real-time applications, psychoacoustic software).

1.2 The team Analyse/Synthèse

The Analysis/Synthesis team of the IRCAM conceives and develops utilities with the aid of which the composers and musicians can transform existing sounds and create new ones.

In order to achieve this, the information transmitted by the sound has to be structured in a clear and intelligible way. This leads to the definition of a model of the sound, which also determins the way of the synthesis. On the one hand, these models explain penomena of sound, and on the other hand, they provide means to modify the sound in a more detailed and precise manner than by global transformations.

In sound synthesis, several models have been worked out and implemented by the team Analyse/Synthése, such as the synthétiseur chant. Every model is being realised in form of a program, called synthétiseur, which calculates the sound from the values of given parameters. This leads to a different view of modifying sound: it is now merely the modifying of the parameters which describe the sound; sound creation in this way means providing new parameters for a new, synthetic sound.

The analysis part of the work now consists in determining the parameters of a given sound. This is a main field of study in this department, and this is also the field which constituted my assignment, the detection of fundamental frequencies.

1.3 Description of my assignment

'Development of an algorithm "Expectation Maximization" for the estimate of several simultaneous fundamental frequencies. Musical signals almost always present several simultaneous fundamental frequencies, even in the case of a monophonic instrument because the reverberation of the room prolongs the previous note on the note in progress. The detection of several simultaneous fundamental frequencies is thus a significant problem. Various work has been done on this subject, in particular in the Analysis-Synthesis team of the IRCAM, but a model and an algorithm satisfactory from the theoretic and practical point of view remains to design and develop.'

Chapter 2

The Algorithm `f02bisd`

2.1 Motivation

In this chapter, the ideas leading to the algorithm will be discussed.

`f02bisd` is an algorithm which is designed to find the fundamental frequency of a given signal of monophonic sound. In a first step, we create models for each f0; then we check how well each of these models describes the actual spectrum of the signal. This is done by basically four summands, `d1` to `d4`. Each of these summands analyzes certain aspects of the spectrum and assigns it a value between 0 and 1. These "aspects" will be described in the following subsections.

2.2 Constructing a model

The spectrum of a sound which we consider as a harmonic sound consists of peaks which are located at a frequency `f0`, called the fundamental frequency, and its integer multiple frequencies, called the superharmonic peaks of `f0`.

We now consider the case where we have an arbitrary windowed signal which we suppose to be harmonic, with a fundamental frequency in the range [f0min, f0max]. Since we don't have any further assumptions or restrictions on the frequencies, every integer frequency is a possible candidate for the f0 of the given sound, and so we construct a model for each integer frequency.

A model for frequency f0 is (an abstraction of) a harmonic spectrum, given by a set of n points of the format $(k \cdot \mathrm{f0}, |S(k \cdot \mathrm{f0})|)_{1 \leq k \leq n}$, where n is the number of peaks in the model ($n = \lfloor \frac{\mathrm{fmax}}{\mathrm{f0}} \rfloor$) and S is the complex spectrum of the windowed signal, evaluated at the frequencies of the multiples of f0.

2.3 Preanalyzing the spectrum of the signal

Before we start comparing our f0-models with the given signal, we can find out some properties of the signal spectrum. The most important property of the spectrum are the frequencies and the amplitudes of its peaks. Hence, the first idea is to detect all peaks of the spectrum, i.e. the maxima of the absolute value of the spectrum, with their corresponding frequency and amplitude.

In addition to the amplitude and frequency of each peak, we can also calculate the reassignment values; a short explanation of this theory will be given LATER.

We now have determined a set $(p_i)_{i=1,\dots,n}$ of spectral peaks, where each peak has a frequency f_i, an amplitude a_i and a reassignment value t_i. It is now possible to select peaks according to their reassignment value. At the current state of the algorithm, there are two ways in which these reassignment values are used: the first is the "transient peak removal", where the peak with the highest amplitude is removed if its reassignment value surpasses a given threshold, i.e. it indicates that the time center of the frequency is too far away from the window center. After the removal of the peak, we can detect other peaks in the errorspectrum, which may have been covered by the bigger peak. The second way is to absolutely ignore those peaks which have "outlier" reassignment values. These peaks are then taken out of the collection (p_i).

2.4 Distance Components

2.4.1 Correlation component - `d1`

`Files:`

- `private/mtlComputeCorrShift5bopt.cpp`
- `private/mtlComputeCorrShift.mexglx (soft link)`

`Functions:`

- `comp-corr()`

This component is no longer in use as standalone component - hence its corresponding parameter `p1` is usually set to 0. Nevertheless, I will explain the purpose of this component, since it has now been combined with the second component, `d2`.

For a given model for frequence f0, we will divide the peaks of the observed spectrum into two groups: those who are "explained" by the model and those who are not.

To the first group, we count all those peaks, who are close enough to one of the modelpeaks, i. e. whose frequency distance to the nearest modelpeak doesn't surpass the threshold $\alpha \cdot f0$. It has been shown in various test series that a good value for the parameter α is 0.4. However, if we don't want all peaks to be explained by the model, we must set α to a value strictly less than 0.5.

The Correlation Component was designed to give an estimation of the "harmonicity" of the observed spectrum. It takes every peak and judges if its amplitude and phase shape is similar to a "modelpeak", i.e. the peak of the FFT of the window.

This is done by a variant of the usual correlation coefficient. Let win denote the model peak amplitude (i.e. the absolute value of the spectrum) and specs_i denote the complex spectrum segment around the spectral peak p_i; both vectors are given in bin values and both have the size of a windowpeak (maximum

$\frac{3 \cdot \text{fftsize}}{\text{windowsize}}$). Then

$$c_i := \frac{|\sum_z \text{win}(z) \cdot \text{specs}_i(z)|}{\sqrt{\sum_z |\text{win}(z)|^2 \cdot \sum_z |\text{specs}_i(z)|^2}}$$

denotes the correlation value of the peak p_i. Some properties of this expression are:

1. $c_i \in [0, 1]$
2. $|\text{specs}_i| = \text{win}$ and specs_i has constant phase $\Longrightarrow$ $c_i = 1$.
3. If specs_i has constant phase and specs_j has the same amplitude, but varying phase, then $c_i \leq c_j$.

Hence a peak "looks good" if its correlation value is close to 1.

2.4.2 Shift component - `d2`

`Files:`

- `private/mtlComputeCorrShift5bopt.cpp`
- `private/mtlComputeCorrShift.mexglx (soft link)`

`Functions:`

- `comp-shift()`

This distance component gives informations about the positions of the observed peaks, relative to the harmonic peaks of a model f0 frequency.

Since this component consists itself of several factors, we will explain each one of them separately.

1. $\vec{\hat{s}}_{f0}$:

   For each peak p_i of the observed spectrum, the distance component $\hat{s}_{f0,i}$ measures the distance to the nearest modelpeak, if f0 explains p_i, i.e. if $|f_i - \text{round}(\frac{f_i}{f0}) \cdot f0| < \alpha \cdot f0$. In this case we define

   $$\hat{s}_{f0,i} := \frac{|f_i - k \cdot f0|}{\alpha \cdot f0},$$

   where $k = \text{round}(\frac{f_i}{f0})$. If f0 does not explain p_i, we set the value of $\hat{s}_{f0,i}$ to 1.

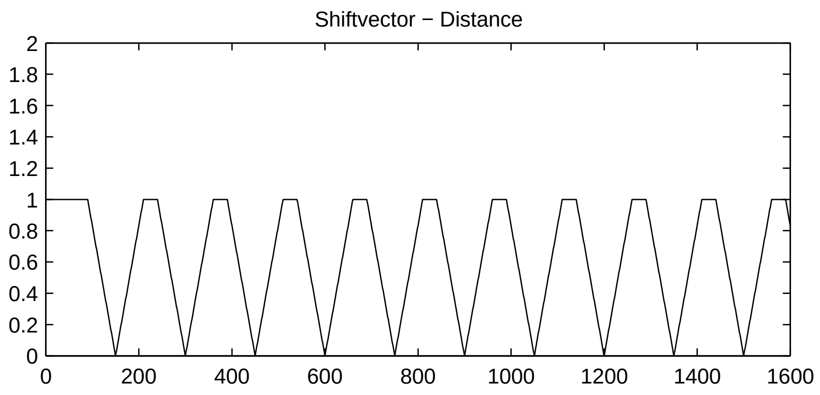
   Properties of this definition:

   (a) If $f_i = k \cdot f0$, then $\hat{s}_{f0,i} = 0$.

   (b) $\hat{s}_{f0,i} \in [0, 1]$.

   (c) If p_i is not explained by f0: $\Longrightarrow$ $\hat{s}_{f0,i} = 1$.

   In other words: If we have a peak p_i, with frequency f_i, then its distance value $\hat{s}_{f0,i}$ is determined according to the following graphic (here, f0 is set to 150 and α to 0.4).



2. c_i:

   This is the Correlation Component as explained in 2.4.1.

3. h_{f0}:

   For each peak p_i of the observed spectrum, the amplitude envelope component $h_{f0,i}$ provides the weight vector for a weighted average on the shift vector. The idea is that only the first n partials of a harmonic spectrum are resolved high enough to give information about the fundamental frequency, and so we want to pay more attention to the lower peaks. Hence, we set

   $$h_{f0,i} := \min(\frac{1}{n \cdot f0}, \frac{1}{f_i}),$$

   where f_i is the frequency of the peak p_i.

Now let $\vec{w}$ be the weighting vector for $\vec{s}$, which is defined (for each bin l) as follows:

$$w_{f0,l} := \begin{cases} c_{f0,i}^{\delta_1} \cdot h_{f0,i}^{\delta_2} & , \quad l \text{ lies in the pondregion of any } p_i \\ 0 & , \quad \text{otherwise.} \end{cases}$$

δ_1 and δ_2 are additional parameters, whose optimal settings are still to determine (see 3).

Let $\vec{\tilde{w}}$ be the normalized version of $\vec{w}$ ($\vec{\tilde{w}} = \frac{1}{\sum_l w_{f0,l} \cdot \text{specs}(l)} \cdot \vec{w}_{f0}$). Then we define the shift vector $\vec{s}$:

$$s_{f0,l} := \begin{cases} \tilde{w}_{f0,l} \cdot \hat{s}_{f0,i} & , \quad l \text{ lies in the pondregion of any } p_i \\ 1 & , \quad \text{otherwise.} \end{cases}$$

2.4.3 Amplitude Envelope component - `d3`

`Files:`

- `private/mtlComputeCorrShift5bopt.cpp`
- `private/mtlComputeCorrShift.mexglx (soft link)`
- `private/newdistbis.m`

`Functions:`

- `comp-shift()`
- `HiLowFrequRatio()`

This component is being added to the other distance summands in order to avoid subharmonic errors; the idea is that the real $f0$ will have a "smoother" amplitude envelope than $\frac{f0}{2}$, i. e. the number of "ups" and "downs" in the amplitude of the observed spectrum, evaluated at the modelpeak frequencies, is lower.

Let $\vec{a}$ be defined as follows:

If p_i is a peak of the observed spectrum and k is the (integer) number of the nearest peak of the model for $f0$. Then, if p_i is explained by $f0$, i.e. if $|f_i - k \cdot f0| < \alpha \cdot f0$, then we define

$$a_k := |\mathrm{specs}(k \cdot f0)|,$$

where specs denotes the observed spectrum. The resulting amplitude vector $\vec{a}$ has then $n = \lfloor \frac{\mathrm{fmax}}{\mathrm{f0}} \rfloor$ entries of the observed amplitude at the frequency of the modelpeaks of $f0$. The entries of $\vec{a}$, whose indices belong to modelpeaks which don't explain any observed peak, are set to 0.

Now define $\vec{\tilde{a}}$ by

$$\vec{\tilde{a}} := (a_n, a_{n-1}, \ldots, a_1, a_1, \ldots, a_{n-1}, a_n),$$

(i.e. $\vec{\tilde{a}}$ is a version of $\vec{a}$, which has been flipped around 0) and let c be a Highpass-Filter.

Then

$$hlr_{f0} := \frac{\sum_{i=1}^{2n} c(\vec{\tilde{a}})_i^2}{\sum_{i=1}^{2n} \tilde{a}_i^2}$$

2.4.4 Reassignment Operator Variation component - `d4`

Even in monophonic signals, effects like echo and hall can result in different f0 frequencies occurring at the same time. In order to avoid confusion between these different sections of the segment, we calculate the reassignment operator r_i for each peak p_i, as described in 2.3.

After having eliminated all those peaks with reassignment operators which indicate an energy center position too far away from the window center (more than 18 percent of the windowsize), we now consider the set $J_{f0} \subseteq \{1, \ldots, n\}$, such that $(p_i)_{i \in J_{f0}}$ are the peaks, which are explained by f0. If r_i is the reassignment operator for p_i and $\hat{s}_{f0,i}$ is as in 2.4.2, then let

$$\hat{w}_{f0,i} := 1 - \hat{s}_{f0,i}$$

and

$$w_{f0,i} := \frac{1}{\sum_{i \in J} \mathrm{specs}(f_j)^2 \cdot \hat{w}_{f0,j}} \cdot \mathrm{specs}(f_i)^2 \cdot \hat{w}_{f0,i}$$

for every $i \in J_{f0}$. This means, $\vec{w}$ is a weighting vector, which moves the "importance" of peaks both towards the model peaks of f0 (we assume that peaks close to harmonic peaks of f0 belong to this position) and towards the peaks of the spectrum with high amplitude (here we also assume that large peaks always carry information about the fundamental frequency).

The variance v_{f0} is then defined as follows:

$$v_{f0} := \sqrt{\sum_{i \in J} (r_i - \sum_{j \in J} w_j \cdot r_j)^2 \cdot w_i}.$$

That means that the only difference to a "regular" variance is that we don't use the arithmetic mean as weight (which would be the case if $w = \frac{1}{n}$), but instead a weighted average, which takes in account the more important peaks.

2.5 Calculating Reassignment operators of the spectrogram

One way to display the energy density of a signal x (in time and frequency) is the Wigner-Ville-distribution (WVD), defined by

$$\mathrm{WVD}(x;t,w) := \int_{-\infty}^{+\infty} x(t+\frac{\tau}{2})x^*(t-\frac{\tau}{2})e^{-i\omega\tau}d\tau.$$

The function

$$\mathrm{WVD}(h;s,\xi)\mathrm{WVD}(x;t-s,\nu-\xi)$$

can be considered as a mass distribution function of the signal x in the window h; hence the time coordinate of the mass center can be found at

$$\hat{t}(x;t,\omega) := t - \frac{\iint u\mathrm{WVD}(h;u,\Omega)\mathrm{WVD}(x;t-u,\omega-\Omega)dud\Omega}{\iint \mathrm{WVD}(h;u,\Omega)\mathrm{WVD}(x;t-u,\omega-\Omega)dud\Omega}. \tag{2.1}$$

It is shown in [1] that this can also be written as

$$\hat{t}(x;t,\omega) = -\frac{d(\arg(\mathrm{STFT}_h(x;t,\nu)))}{d\nu}, \tag{2.2}$$

which means that the reassignment values depend strongly on the phase of the Fouriertransform of x, not only on the form of the amplitude. As we will show now, $\hat{t}$ is equal to the expression

$$t - \mathrm{Re}\left(\frac{\mathrm{STFT}_{Th}(x;t,\omega)\mathrm{STFT}_h(x;t,\omega)^*}{|\mathrm{STFT}_h(x;t,\omega)|^2}\right), \tag{2.3}$$

where $Th(s) = s \cdot h(s)$ is a modified window. Since this is a faster and numerically more stable version of (2.1), the reassignment values are now used in `f02bisd` in the form (2.3). Remains to show the equality

$$\frac{\frac{1}{2\pi}\iint u\mathrm{WVD}(h;u,\Omega)\mathrm{WVD}(x;t-u,\omega-\Omega)dud\Omega}{\frac{1}{2\pi}\iint \mathrm{WVD}(h;u,\Omega)\mathrm{WVD}(x;t-u,\omega-\Omega)dud\Omega} = \mathrm{Re}\left(\frac{\mathrm{STFT}_{Th}(x;t,\omega)\mathrm{STFT}_h(x;t,\omega)^*}{|\mathrm{STFT}_h(x;t,\omega)|^2}\right). \tag{2.4}$$

We will only show the equality of the numerators - the equality of the denominators follows with a similar argument.

$$
\begin{aligned}
& \frac{1}{2\pi}\iint u\mathrm{WVD}(h;u,\Omega)\mathrm{WVD}(x;t-u,\omega-\Omega)dud\Omega \\
= \quad & \frac{1}{2\pi}\iint u\int h(u+\frac{\tau}{2})h^*(u-\frac{\tau}{2})e^{-i\Omega\tau}d\tau\int x(t-u+\frac{\rho}{2})x^*(t-u-\frac{\rho}{2})e^{-i(\omega-\Omega)}d\rho dud\Omega \\
= \quad & \iint ux(t-u+\frac{\rho}{2})x^*(t-u-\frac{\rho}{2})e^{-i\omega\rho}\frac{1}{2\pi}\iint h(u+\frac{\sigma+\rho}{2})h^*(u-\frac{\sigma+\rho}{2})e^{-i\Omega\sigma}d\sigma d\Omega d\rho du,
\end{aligned}
$$

with $\sigma := \tau-\rho$. Setting $z(\sigma) := h(u+\frac{\sigma+\rho}{2})h^*(u-\frac{\sigma+\rho}{2})$, this yields

$$
\begin{aligned}
= \quad & \iint ux(t-u+\frac{\rho}{2})x^*(t-u-\frac{\rho}{2})e^{-i\omega\rho}\frac{1}{2\pi}\int Z(i\Omega)d\Omega d\rho du \\
= \quad & \iint ux(t-u+\frac{\rho}{2})x^*(t-u-\frac{\rho}{2})e^{-i\omega\rho}z(0)d\rho du; \\
= \quad & \iint ux(t-u+\frac{\rho}{2})x^*(t-u-\frac{\rho}{2})e^{-i\omega\rho}h(u+\frac{\rho}{2})h^*(u-\frac{\rho}{2})d\rho du;
\end{aligned}
$$

where $Z(i\Omega)$ is the Fourier Transform of z. Now set $v := t-u+\frac{\rho}{2}$:

$$
\begin{aligned}
= \quad & -\iint(t-v+\frac{\rho}{2})x(v)x^*(v-\rho)e^{-i\omega\rho}h(t-(v-\rho))h^*(t-v)d\rho dv \\
= \quad & -\iint\left[x(v)(t-v)^*h^*(t-v)e^{-i\omega v}\right]\left[x(v-\rho)h^*(t-(v-\rho))e^{-i\omega(v-\rho)}\right]^* d\rho dv \\
- \quad & \iint\underbrace{\frac{\rho}{2}x(v)x^*(v-\rho)e^{-i\omega\rho}h(t-(v-\rho))h^*(t-v)}_{:=s(\rho)}d\rho dv \\
= \quad & \mathrm{STFT}_{Th}(x;t,\omega)\mathrm{STFT}_h(x;t,\omega)^* - \iint s(\rho)d\rho dv.
\end{aligned}
$$

The equality 2.4 now follows from the fact that WVD is always real (for symmetry reasons: $x(u+\frac{\tau}{2})x^*(u-\frac{\tau}{2})e^{-i\tau\omega} = \left[x(u-\frac{\tau}{2})x^*(u+\frac{\tau}{2})e^{-i(-\tau)\omega}\right]^*$) whereas the subtrahend of the last line is purely imaginary (also by symmetry: $s(\rho) = -s(-\rho)^*$).

Chapter 3

Testing the algorithm

3.1 Test procedures

Minimizing the error rate of an f0-algorithm, applied to monophonic signals, requires both a representative choice of sound files and an exact idea of the "correct" f0-values. In our case, the chosen signals were speech signals of male and female speakers. Following the example of A. Cheveigné ([3]), we used the soundfiles from the Bagshaw database (Bagshaw et al, 1993, http://www.cstr.ed.ac.uk/~pcb/fda_eval.tar.gz - or on the IRCAM network under /net/lilith/alain/f0data/fda) to evaluate and optimize the algorithm. The sound files in this database were recorded together with the signal of a laryngograph (an apparatus that measures electrical resistance between electrodes placed across the larynx), from which a reliable "ground-truth"can be derived.

`f02bisd` was called with f0min = 40 Hz, f0max = 800 Hz, a windowsize corresponding to 40 ms and a blockshift (distance between the centers of two windows) corresponding to 2,5 ms.

When evaluating the results, values that differed by more than 20% from the laryngograph-derived estimates were counted as "gross errors". These errors can further be divided into "too low" (mainly subharmonic) and "too high" errors. The resulting error rate is the number of errors, divided by the number of compared results.

3.2 Tests with two parameters

Since the beginning of my internship, various attempts have been made to improve the results of the given algorithm, `f02bisd`, which had been worked out by B. Prudham [8]. Namely the first component, (1-corr), which seemed to produce results which are only vaguely connected to the appearance of the spectrum, due to the strongly oscillating aspect of (1-corr).

Furthermore, the effects of the exponentiation used ($\nu_{\text{expo}}(k)$, as defined in [8]), are not evident. Hence we removed the summand `d1` by setting `p1` to 0, and used the correlation vector as a weighting vector for the second component only; for details, see 2.4.2.

After having introduced two new parameters (δ_1 and δ_2, as in 2.4.1), we tested the algorithm `f02bisd` in various parameter settings in order to study how the parameters would be best adjusted, in interaction with the "old" parameters `p1` to `p3`. We assigned a range (e.g. 1.0:0.1:2.0 in `testcoeffscrexp3.m`), another one to `d2` (e.g. 6:4:42) and tested the algorithm with every possible combination of the two parameter settings, while fixing the other parameters. The preceding example would result in 110 calls of the function `f02bisd.m`.

The following table 3.1 gives an overview of the scripts which were used to do tests with two parameters.

Scriptfile	Parameters adjusted	Soundfile
`testcoeffscrexp3.m`	`distexp2, d2`	`rl001`
`testcoeffscrexp4.m`	`distexp, d2`	`rl002`
`testcoeffscrexp5.m`	`distexp1, distexp2`	`rl001`
`testcoeffscrexp0_5.m`	`distexp, d2`	`rl001`
`testcoeffscrexp0_75.m`	`distexp, d2`	`rl001`
`testcoeffscrexp.m`	`distexp, d2`	`rl001`
`testcoeffscrexpalpha.m`	`alpha, d2`	`rl001`
`test2coeffscr.m`	`distexp, d2`	`sb001`
`test2coeffscr2.m`	`distexp1, distexp2`	`sb001`

Table 3.1: Test directories

Test results can be displayed (divided into sub - and superharmonic errors) with scripts like `displaysubsup4.m`. Since these results do not reflect the current state of the algorithm, we will abstain from presenting them here in detail.

3.3 Tests with three parameters

After having introduced a second exponent `distexp2` (the one which is in 2.4.1 referred to as $\delta 2$), which is intended to put more weight on the lower regions of the signal (since it increases the amplitude envelope at the frequences lower than f0), we have now no longer the problem of testing two, but three different parameters for the algorithm. This leads to three arrays of possible values, where every triple of values has to be tested.

Except for the difficulty to display the test results intuitively, this test scheme poses also problems in terms of calculation time: if every parameter test array is of length 10, we already have 1000 calls of `f02bisd.m`, which is equivalent to roughly 80 hours of calculation on one machine. This shows the need for more efficient and time-saving test algorithms.

Scriptfile	Parameters adjusted	Soundfile
`testexp3d.m`	`distexp1, distexp2, d2`	`rl001`
`testexp3d1.m`	`distexp1, distexp2, d2`	`rl001`
`test2coeffscr3d.m`	`distexp1, distexp2, d2`	`sb001`
`test2coeffscr3d1.m`	`distexp1, distexp2, d2`	`sb001`

Table 3.2: Test directories

The results can be displayed with scripts like `displaysubsup13d.m`.

3.4 Evolutionary test algorithm

Having introduced yet another parameter (the standard deviation `d4`), the method of testing every combination of parameters became nearly impossible, due to the huge amount of time this would have taken. Instead, we implemented an algorithm which combines evolutionary and genetic strategies to find the "best" adaptation of parameters.

Evolution Strategy (ES) was developed at Berlin Technical University by Ingo Rechenberg (Rechenberg 1973) and Hans Peter Schwefel (Schwefel 1981). The ES-algorithms consider the indivual i.e. its phenotype to be the object to be optimized.

The characteristical data of the individual are the parameters to be optimized in a evolution-based process. These parameters are arranged in vectors of real numbers on whom operators for recombination (cross-over) and mutation are defined.

A population is represented by n individuals $P = (c_1, \ldots, c_n)$, where $c_i = (p_{i,1}, \ldots, p_{i,m})$. $p_{i,j}$ is thus the jth object parameter of the ith individual c_i, and is in fact a real number which constitutes one of the parameter settings for the algorithm `f02bisd.m`.

The evolutionary process starts with a randomly chosen start generation P_0. If P_l is already defined, we get P_{l+1} by applying the three steps Recombination, Mutation and Selection. This process can then be repeated, until the parameters in the current generation are sufficiently adapted.

1. Recombination step

   Let $i \in \{0, \ldots, \lfloor \frac{n}{2} \rfloor - 1\}$, and let π be a random permutation of the indizes of the parent generation. We select two "parent individuals" $c_{\pi(2i+1)}$ and c_{2i+2} and construct the new individual c_{n+i} by choosing each of its object parameters randomly from one of the parents, i.e. $c_{n+i} = (p_{k_1,1}, \ldots, p_{k_m,m})$, where

   $$k_j = \begin{cases} \pi(2i+1) & , \quad \text{rand}_j < .5 \\ \pi(2i+2) & , \quad \text{otherwise,} \end{cases}$$

   where rand_j is an equally distributed random number on [0,1].

2. Mutation step

   The second way to generate new individuals from a given population is mutation. In our implementation, we have two mutated individuals per parent; i.e. if $i \in \{1, \ldots, n\}$, then the new individuals are $c_{n+\lfloor \frac{n}{2} \rfloor+2i-1}$ and $c_{n+\lfloor \frac{n}{2} \rfloor+2i}$, with $c_{n+\lfloor \frac{n}{2} \rfloor+2i-1} = (p_{k_1}, \ldots, p_{k_m})$ and

   $$p_{k_j} = \begin{cases} p_{i,j} + \text{randn}_j & , \quad \text{rand}_j < .2 \\ \text{rand}'_j & , \quad \text{otherwise;} \end{cases}$$

   In this case, randn_j denotes a normally distributed random variable, rand_j denotes an equally distributed random number on [0,1] and rand'_j is an equally distributed random number on the parameter range of the jth object parameter.

3. Selection step Now, $P'_l = (c_1, \ldots, c_n, c_{n+1}, \ldots, c_{n+\lfloor\frac{n}{2}\rfloor}, c_{n+\lfloor\frac{n}{2}\rfloor+1}, \ldots, c_{n+\lfloor\frac{n}{2}\rfloor+2n})$ is the current generation, consisting of n parent individuals $\lfloor\frac{n}{2}\rfloor$ recombinated and $2n$ mutated individuals. We now apply the algorithm `f02bisd.m` with the appropriate object parameters of each individual c_i and calculate the error quote q_i.

   Then P_l consists of the n individuals with the lowest error quotes. In this way, we always move towards a lower overall error quote in every generation.

The following table 3.3 shows the directories used to store results of this evolutionary strategy. All directories which are listed here used at least the files `rl010` and `sb004` from the database of Bagshaw, and all are located in `/net/kecer/data1/krauleda/`. The backup directories are in `/u/formes/krauleda/`.

Directory	Comments/changes	Backup	Result
`generictest/`			9.43%
`genetictest2/`	introduced post processing (cross out too high decalations of estimates)		8.40%
`genetictest4/`	`d2 = d2`^{2}; Bug fixed: NaN in variance		8.61%
`genetictest5/`	`d2 = d2`^{1}		8.82%
`genetictest6/`	`d2 = d2`^{2}; increased training dataset by `rl001`	`backup2/`	7.50%
`genetictest8/`	New parameter: NumParts; increased training dataset by `sb011`		10.29%
`genetictest9/`	Re-introduced `d1; d2 = d2`^{2}	`backup6/`	7.69%
`genetictest10/`	`d2 = d2`^{1}	`backup3/`	8.69%
`genetictest11/`	Restricted minimum search to local minima of `d2`.	`backup4/`	8.49%
`genetictest12/`	Re-introduced varying explanation region	`backup5/`	9.63%

Table 3.3: Evolutionary tests

Most of the scripts with which these tests were run have the same names as the corresponding folders. The column "Result" contains the lowest error percentage of the latest generation of parameters. Since we picked extremely difficult signals for training purposes, these percentage doesn't reflect the actual performance of the algorithm.

3.5 Entire Database tests

Having found some promising parameter settings with the above methods, we started test runs on the entire database of Bagshaw. The directories with the results are listed in table 3.4.

Directory	Parameters	Version	Result
`testdatabase/`	$\delta_1 = 1.75, \delta_2 = 1.6$,p2= 14		2.27%
`testdatabase2/`	$\delta_1 = 2.0, \delta_2 = 1.7$,p2= 26		2.55%
`testdatabase3/`	$\delta_1 = 0, \delta_2 = 3.5$,coeff $= 0, 30, 15, 16$	`backup/`	2.66%
`testdatabase4/`	$\delta_1 = 0.5, \delta_2 = 4.5$,coeff $= 16, 10, 15, 10$	`backup6/`	2.10%
`testdatabase5/`	$\delta_1 = 0.5, \delta_2 = 4.5$,coeff $= 16, 10, 15, 10$ (without discarding transient peaks)	`backup7/`	2.04%

Table 3.4: Database tests

The column "Result" gives the overall result of the test on the database. It is the proportion of the number of errors to the number of total tests.

More detailed results can be displayed by using the script `displaydatabase`.

In comparison with the old version of the algorithm, we can remark that the latest verion of the algorithm, stored in `backup7/`, performs slightly better on the Bagshaw database (2.04% vs. 2.15%, as displayed in table 3.6 and 3.5).

The percentage given in table 3.5 differs from the one calculated in [8]; this is due to the different calculation method employed: B. Prudham has calculated the error rate for every file in the database and has then taken the arithmetic mean over all percentage values, whereas we calculated the error rate of all files at once by dividing the number of errors by the number of tests. This approach avoids the influence of the file lengths of the files in the database.

	male	female	all files
sub	0.53%	2.29%	1.50%
sup	0.70%	0.62%	0.67%
Σ	1.23%	2.91%	2.15%

Table 3.5: Prudhams test results in /net/keprak/data2/prudham/testdatabase/

	male	female	all files
sub	0.46%	1.82%	1.21%
sup	0.82%	0.81%	0.82%
Σ	1.29%	2.64%	2.04%

Table 3.6: Test results in testdatabase5/

Compared to the results of YIN on the same database, this is also a slight improvement: according to [3], YIN has a total error rate of 2.2% on the Bagshaw database.

Bibliography

[1] F. AUGER and P. FLANDRIN, *Improving the readability of time-frequency and time-scale representations by the reassignment method*; IEEE transactions on signal processing, vol. 43, no. 5, May 1995.

[2] F. AUGER and P. FLANDRIN, *Time-Frequency Toolbox Tutorial*, CNRS, 1995-96.

[3] A. CHEVEIGNE and H. KAWAHARA, *YIN - A fundamental frequency estimator for speech and music*; Journal of the Acoustical Society of America, 100(4), 1917-1930, April 2002.

[4] A. CHEVEIGNE and A. BASKIND, *F0 estimation of one or several voices*; IRCAM - internal report.

[5] M. DURAND, *Estimation de la fréquence fondamentale et cas de fréquences fondamentales multiples*, IRCAM - internal report, February 2002.

[6] A. V. OPPENHEIM and R. W. SCHAFER, *Discrete-time signal processing*, Prentice Hall Signal Processing Series, 1989.

[7] S. J. ORFANDIS, *Introduction to signal processing*, Prentice Hall international editions, 1996.

[8] B. PRUDHAM, *Estimation de la fréquence fondamentale d'un signal*, IRCAM - internal report, August 2002.

[9] R. UNBEHAUEN, *Systemtheorie - Eine Darstellung für Ingenieure*, Oldenbourg, München, Wien, 1983.