(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) EP 1 437 711 A1

(12) EUROPEAN PATENT APPLICATION

(43) Date of publication:
14.07.2004 Bulletin 2004/29

(51) Int Cl.^{7}: **G10H 1/00**

(21) Application number: **02293122.4**

(22) Date of filing: **17.12.2002**

(84) Designated Contracting States:
AT BE BG CH CY CZ DE DK EE ES FI FR GB GR IE IT LI LU MC NL PT SE SI SK TR
Designated Extension States:
AL LT LV MK RO

(71) Applicant: **Sony France S.A.**
92110 Clichy (FR)

(72) Inventors:
- **Pachet, François**
  75010 Paris (FR)
- **Zils, Aymeric**
  75005 Paris (FR)

(74) Representative: **Bertrand, Didier et al**
S.A. FEDIT-LORIOT et autres
Conseils en Propriété Industrielle,
38, avenue Hoche
75008 Paris (FR)

(54) Method and apparatus for generating a function to extract a global characteristic value of a signal contents

(57) The invention enables to generate a general function (4) which can operate on an input signal (Sx) to extract therefrom a predetermined global characteristic value (DVex) expressing a feature of the information (De) conveyed by that signal.

It operates by:

- generating automatically compound functions (CF1- CFn), each compound function being composed of at least one of a set of elementary functions (EF1, EF2, ..), by using means (22, 25) that handle said elementary functions as a symbolic object, and
- operating said compound functions on at least one reference signal (S1-Sm) having a pre-attributed global characteristic value (Dgt1-Dgtm) serving for evaluation, by using means (22, 27) that process said elementary functions as executable operators,
- determining the correlation between the values (Dij) extracted by those compound functions as a result of operating on said reference signal and the pre-attributed global characteristic value (Dgt1-Dgtm) of said reference signal, and
- selecting said general extraction function (4) among those compound functions for which said correlation is relatively high.

The invention can be used, for instance, for the automatic extraction of audio/music descriptors from their signals contained as music file data.

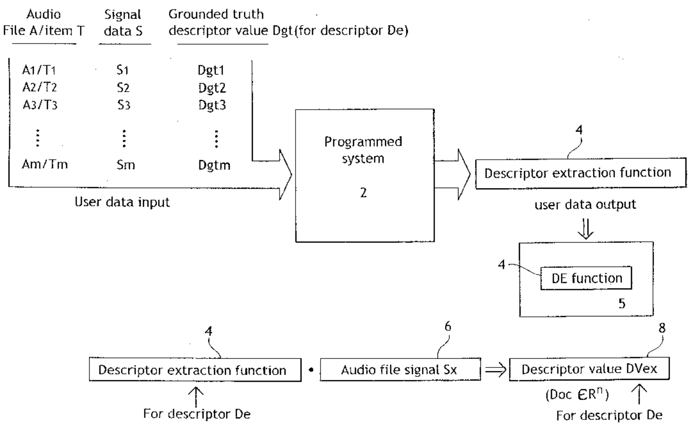


FIG.1

EP 1 437 711 A1

Printed by Jouve, 75001 PARIS (FR)

Description

[0001] The invention relates to the field of signal processing, and more particularly to a technique for deriving automatically high level information expressed by an electronic input signal by analysing the signal's low-level characteristics. In this context, the term high-level refers to the global characteristics of the signal content, while the term low-level refers to the fine grain structure of the signal itself, typically at the level of its temporal or spatial modulation.

[0002] For instance, in the case of audio signals corresponding a given music title, such as contained in an audio file readable by a music player, examples of its high-level expression would be an indication of whether the title pertains to a sung or instrumental piece of music, the musical genre, musical complexity, overall timbre, tempo, or the rhythm structure, etc., while the low-level characteristics would be the signal's time-dependent parameters such as amplitude, pitch, etc. analysed over successive short sampling periods. The signals in question can thus be in the form of digital data accessed from a memory or inputted as a digital stream, or they can be in analogue form.

[0003] In such audio applications, the high-level information is normally known by the term "descriptor". Generally, a descriptor expresses a quality, or dimension, of the content represented by the signal, and which is meaningful to a human or to a machine for processing high-level information. Depending on what they express, descriptors attribute a value which can be of different types:

- a Boolean, e.g. true/false to indicate whether or not a music title is sung,
- a number to express information quantitatively against a reference scale, e.g. 7.3 against a scale of 1 to 10 for a global music energy descriptor,
- an indication of a selection from a list of labels, e.g. "military music" to indicate a musical genre from a preset list.

[0004] In the field of music, descriptors are of interest notably in the expanding field of music access systems and Electronic Music Distribution (EMD). To facilitate user access to large music databases, descriptors of music titles are needed. EMD belongs to the more general concept of music information retrieval (MIR), which is the technique of intelligently searching and accessing musical information in large music databases.

[0005] Traditionally, EMD systems use either manually entered descriptors (e.g. using software systems developed commercially by the companies "Moodlogic" and "AllMusicGuide". The descriptors are then used for accessing music browsers, using a search by similarity, or a search by example, or any other known database searching technique.

[0006] A key issue in extracting automatically descriptors for audio signals is that it is very difficult to map signal properties with perceptive categories. In the prior art, attempts have been made to extract specific descriptors from a sound signal, these being documented notably in:

- Scheirer, Eric D., "*Tempo and Beat Analysis of Acoustic Musical Signals*", J. Acoust. Soc. Am. (JASA) 103:1 (Jan 1998), pp 588-601., for tempo,
- Aucouturier Jean-Julien, Pachet Francois, *"Music Similarity Measures: What's the Use?* ", Proceedings of the 3rd International Symposium on Music Information Retrieval (ISMIR02), Paris - France, October 2002, for timbre,
- Pachet, F., Delerue, O. ,Gouyon, F., "*Extracting Rhythm from Audio Signals ",* SONY Research Forum, Tokyo, December 2000, for rhythm, and.
- Berenzweig A.L., Ellis D. P. W., *"Locating Singing Voice Segments within Music Signals",* IEEE Workshop on Applications of Signal Processing to Acoustics and Audio (WASPAA01), Mohonk NY, October 2001.

[0007] There are however many other dimensions, i.e. descriptors, of music that can be extracted from the signal. For instance:

danceability
music for children
military music
music for slow
global energy
sung versus instrumental
original versus remix
acoustic versus electr(on)ic
live versus studio
musical complexity
musical density
etc.

[0008] While such descriptors are readily discernible by a human listener, the technical problem of producing them electronically from raw music data signals is reputed to be particularly difficult. For instance, there is no immediately apparent low-level characteristic of a raw music signal from which it is possible to identify whether it pertains to a sung piece or to an instrumental. This is particularly true when the sung voice is mixed with music. Even the global energy descriptor has no straightforward link with the energy level of the raw signal.

[0009] Some descriptors, such as the musical genre, are influenced by cultural references and therefore require criteria to be entered from a specific population sample.

[0010] In view of the foregoing, the invention provides for an automated tool which takes for input a test database containing a set of reference signals, for instance audio files readable by a music player, at least one arbitrary descriptor that can be potentially correlated to the signals, a grounded truth value of that descriptor for each of the database signals and a set of elementary signal processing functions. The tool then selects functions of that set to construct one compound function or more, and automatically applies it on the signals of the database. Depending the correlations between the value returned by the function and grounded truths, new compound functions are created and tried, until an arbitrary end condition is reached.

[0011] More particularly, according to a first aspect, the present invention relates to a method of generating a general extraction function which can operate on an input signal to extract therefrom a predetermined global characteristic value expressing a feature of the information conveyed by that signal,

characterised in that it comprises the steps of:

- generating automatically compound functions, each compound function being composed of at least one of a set of elementary functions, by using means that handle the elementary functions as symbolic objects,
- operating said compound functions on at least one reference signal having a pre-attributed global characteristic value and serving for evaluation, by using means that process the elementary functions as executable operators,
- determining the correlation between the values extracted by those compound functions as a result of operating on the reference signal and the pre-attributed global characteristic value of the reference signal, and
- selecting the general extraction function among those compound functions for which the correlation is relatively high.

[0012] The invention provides for many advantageous optional embodiments, which are outlined below.

[0013] The compound functions are preferably generated in successive populations,

wherein each new population of functions takes as a basis earlier population functions which produce a relatively high correlation.

[0014] The method can be performed by the steps of:

a) preparing at least one reference signal for which the predetermined global characteristic value is pre-attributed,
b) preparing a population of compound functions each composed of at least one elementary function,
c) modifying compound functions of the current population using the means that handle their elementary functions as symbolic objects,
d) operating the compound functions of the population on at least one reference signal using the means that exploit the elementary functions as executable operators, to obtain a calculated value for each compound function of the population in respect of the reference signal,
e) for at least some compound functions of the population, determining the degree of matching between its calculated value and the pre-attributed value for the signal from which that value has been calculated,
f) selecting compound functions of the population producing the best matches to form a new population of functions,
g) if an ending criterion is not satisfied, returning to step c), where the new population becomes the current population,
h) if an ending criterion is satisfied, outputting at least one compound function of the current new population as a general function.

[0015] The compound functions are preferably produced by random choices guided by rules and/or heuristics.

[0016] The rules and/or heuristics can comprise at least one rule which forbids, from a random draw for selecting an elementary function to be associated with a part of a compound function under construction, an elementary function that would be formally inappropriate for that part.

[0017] The rules and/or heuristics can comprise at least one heuristic which favours, in a random draw for selecting an elementary function to be associated with a part of a compound function under construction, an elementary function which is considered to produce potentially useful technical effects in association with that part, and/or which discourages from the random draw an elementary function considered to produce technical effects of little or no use in association with that part.

[0018] The rules and/or heuristics can comprise at least one heuristic which ensures that a compound function comprises only elementary functions that each produce a meaningful technical effect in their context.

[0019] The rules and/or heuristics can comprise at least one heuristic which takes into account at least one overall characteristic of the reference signals.

[0020] Advantageously, a new population of functions is produced using genetic programming techniques.

[0021] The genetic programming techniques comprise at least one of following:

- crossover,
- mutation,
- cloning.

[0022] A crossover operation and/or a mutation operation can be guided by at least one heuristic cited above.

[0023] The means that handle the elementary functions as symbolic objects preferably manage the functions in accordance with a tree structure comprising nodes and connecting branches, in which each node corresponds to a symbolic representation of a constituent unit function, the tree having a topography in accordance with the structure of the function.

[0024] Advantageously, the method further comprises a step of submitting a compound function to at least one rewriting rule executed by processing means to ensure that said compound function is cast in its most rational form or most efficient form in respect of execution efficiency.

[0025] Preferably the method uses a caching technique for evaluating a function, in which results of previously calculated parts of functions are stored in correspondence with those parts, and a function currently under calculation is initially analysed to determine whether at least a part of said function can be replaced by a corresponding stored result, said part being replaced by its corresponding result if such is the case.

[0026] The method can then comprise the steps of checking the usefulness of results stored according to a determined criterion, and of erasing those found not to be useful, the criterion for keeping a result Ri being a function which takes into account: i) the calculation time to produce Ri, ii) the frequency of use of Ri and, optionally, iii) the size (in bytes) of Ri.

[0027] The elementary functions can comprise signal processing operators and mathematical operators.

[0028] The method can further comprise a step of validating a general function against at least one reference signal having a known value for the general characteristic, and which was not used to serve as the reference.

[0029] The signal can express an audio content, and the global characteristic can be a descriptor of the audio content.

[0030] The audio content can be in the form of an audio file, the signal being the signal data of the file.

[0031] Examples of descriptors for which the invention can be use are:

- a global energy indication,
- a sung or instrumental audio content,
- an evaluation of the danceability,
- an acoustic or electric sounding audio content,
- presence or absence of a solo instrument, e.g. guitar or saxophone solo.

[0032] According to a second aspect, the invention relates to a method of extracting a global characteristic value expressing a feature of the information conveyed by a signal, characterised in that it comprises calculating for that signal the value of a general function produced specifically by the method according to the first aspect for that global characteristic.

[0033] According to a third aspect, the invention relates an apparatus for generating a general function which can operate on an input signal to extract therefrom a predetermined global characteristic value expressing a feature of the information conveyed by that signal,

characterised in that it comprises:

- means for generating automatically compound functions, each compound function being composed of at least one of a set of elementary functions, the means handling the elementary functions as symbolic objects,
- means for operating the compound functions on at least one reference signal having a pre-attributed global characteristic value serving for evaluation, the means processing the elementary functions as executable operators,
- means for determining the correlation between the values extracted by those compound functions as a result of operating on the reference signal and the pre-attributed global characteristic value of the reference signal, and
- means for selecting the general extraction function among those compound functions for which the correlation is relatively high.

[0034] According to a third aspect, the invention relates to an apparatus according to the third aspect configured to execute any one of the optional aspects of the method set out above, it being understood that the features defined in the context of the method can be implemented mutatis mutandis to the apparatus.

[0035] According to an fourth aspect, the invention relates to the use of the apparatus according to the third aspect as a fully autonomous automatic descriptor extraction function generating system.

[0036] According to a fifth aspect, the invention relates to the use of the apparatus according to the third aspect as a descriptor extraction means.

[0037] According to a sixth aspect, the invention relates to the use of the apparatus according to the third aspect as an authoring tool for producing descriptor extraction functions.

[0038] According to a seventh aspect, the invention relates to the use of the apparatus according to the third aspect as an evaluation tool for externally produced descriptor extraction functions.

[0039] According to an eighth aspect, the invention relates to a general function in a form exploitable by an electronic machine, produced specifically by the apparatus according to the third aspect.

[0040] According to a ninth aspect, the invention relates to a software product containing executable code which, when loaded in a data processing apparatus, enables the latter to perform the method according to the first aspect.

[0041] In the preferred embodiment, the above iterative search procedure through successive populations is implemented by what is known as genetic programming. The functions — which typically take the form of executable code — are tried and the results serve to automatically create new populations of functions in accordance with genetic programming techniques, taking the best fitting functions in a manner somewhat analogous to selection and submitting those selected functions to actions corresponding e.g. to crossover and mutation phenomena occurring in biological processes at chromosome level. The remarkable aspect here resides in applying a genetic programming technique on functions which take for argument raw electronic signals.

[0042] When applied to the field of music files, the proposed invention allows to extract arbitrary descriptors from music signals. More precisely, the embodiment does not extract a particular descriptor, but rather, given a set of music titles containing both examples (and possibly counter-examples) for a given descriptor, builds automatically a function that extracts from audio signals an optimum value. The same system can be used to produce a function associated to an arbitrary descriptor such as one listed in the earlier part of the introduction, which can then be exploited as a general function for that associated descriptor, in the sense that it can be made to operate subsequently on any music file to extract the value of the descriptor for that file (assuming its signals are compatible).

[0043] The design of the system is based on extended experiments in the field of audio/music description extraction. During these experiments the applicant observed that a deep knowledge of signal processing was required to design accurate and robust signal processing extractors. Each extractor can be seen here as a function that takes as argument a given music signal (typically 3 minutes of audio), and outputs a value. This value can be of various types: a float (for the tempo), a vector (for the timbre), a symbol (for instrumental versus song discrimination), etc.

[0044] The main task of extractor design is to find the right composition of basic, low-level signal processing functions to yield a value that is as correlated as possible to the values obtained by psycho-acoustic tests.

[0045] The preferred embodiment contains a representation of a human expertise in signal processing: it will try different combinations of signal processing functions, evaluate them, and compare them against human perceptive values. Using an algorithm based on genetic programming, different signal processing functions will be tried concurrently, and modified to find a satisfying extractor function.

[0046] Compared to existing approaches in music extraction, the system is one step higher: its primary function is not to produce a descriptor for a signal, but rather a function which itself will produce the descriptor, when applied on other music file signals e.g. taken from a database of signals.

[0047] The invention and its advantages shall become more apparent from reading the following description of the preferred embodiments, given purely as nonlimiting examples, with reference to the appended drawings in which:

- figure 1 is a diagram showing the basic user input and output of a programmed system for automatically generating descriptor extraction functions in accordance with the invention;
- figure 2 is a simplified block diagram showing the main functional units of the system shown in figure 1;
- figure 3 is a symbolic illustration showing the formal compatibility requirements for two grouped elementary functions forming part of a compound function produced by the system of figure 2;
- figure 4 is a symbolic illustration of an elementary function for performing a low-pass filtering operation on a signal;
- figure 5 is a symbolic illustration of an elementary function for performing a short-time fast Fourier transform operation on a signal;
- figure 6 is a symbolic illustration of a grouping of elementary functions forming a term in a compound function;
- figure 7 is a diagram showing an example of a tree structure symbolic representation of a compound function;
- figure 8 is a diagram showing a matrix of values calculated on a set of reference signals for a population of compound functions, and how those values are used to determine the fit of those functions with respect to a descriptor

associated with the music contents of those signals;

- figure 9 is a diagram showing, through a tree structure representation, how parts of two compound functions are combined to form a new compound function using a crossover operation according to a genetic programming technique;
- figure 10 is a diagram showing, through a tree structure representation, how a compound function is mutated into a new compound function using a mutation operation according to a genetic programming technique;
- figure 11 is a diagram showing, through a tree structure representation, how a caching technique is implemented to acquire results data for a prior-results data cache and to substitute a part of a function under calculation with a previously calculated result;
- figure 12 is a flow chart showing the general steps performed by the system of figure 2 for producing a descriptor extraction function;
- figure 13 is an example of different functions and their fitness produced automatically by the system of figure 2 for evaluating the presence of voice in music title; and
- figure 14 is an example of different compositions of descriptor extraction functions in terms of elementary functions, and their fitness produced automatically by the system to evaluate the global energy of music titles.

[0048] Figure 1 depicts a system 2 in accordance with the invention to indicate the raw data on which it operates (user data input) and the output (user data output) it produces from the latter. The example is based on a music data application, in which the system 2 generates as its user data output an executable function 4, referred to as a descriptor extractor function (DE function). This function is then packaged in a data carrier 5 in a form suitable to be exploited for extracting a given descriptor from an arbitrary audio file 6. The latter is typically formatted according to a recognised standard such as CD audio, MP3, MPEG7, WAV, etc exploitable by a music player, and contains a musical piece to which a descriptor value Dx is to be associated. The DE function 4 operates on the raw data signal Sx of the audio file 6, i.e. it takes the latter as its argument or operand and returns the descriptor value DVex for that file. Naturally, the signal Sx is assumed to be compatible with the DE function 4 as regards data format. As mentioned in the introductory portion, the descriptor value is typically a number, a Boolean, or a statement, and generally belongs to the class or real objects R^n.

[0049] The above data carrier 5 typically comprises a software package which can contain other DE functions, e.g. for extracting other descriptor values, and possibly auxiliary software code, e.g. for management and user assistance. The data carrier 5 can be a physical entity, such as a CD ROM, or it can be in immaterial form, e.g. as downloadable software accessible from the Internet.

[0050] The system 2 generates the DE function 4 on the basis of both the user data input and internally programmed parameters, functions and algorithms, as shall be detailed later.

[0051] The user data input serves inter alia to feed an internal learning database and constitutes the raw learning material from which to model the DE function. This material includes a set of m audio files A1 to Am and, for each one Ai(1 i m), a given value Dgti of a specific descriptor De for the audio item Ti it contains. The audio files Ai are formatted as for file 6 above, and thus each produce a respective signal Si when accessed to reproduce the audio item Ti.

[0052] The respective descriptor values Dgt1-Dgtm associated to the audio files are established by a human judge, or a panel of human judges. For instance, if the descriptor De in question is the "global energy" of the music title, the judge or panel awards for each respective title Ti a number within a range from a minimum (level of a lullaby, for instance) to a maximum, and which constitutes the title's descriptor value Dgti. These values Dgti are referred to "grounded truth" descriptor values.

[0053] Figure 2 shows the general architecture of the system 2. The system is preferably implemented using the hardware of a standard personal computer PC. For ease of understanding, the different types of data used are divided into respective databases 10-18 under the general control of a data management unit 20, which further manages the overall data flow of the system 2. The databases comprise:

- a learning database 10, which stores the signal data S1-Sm of the reference audio files A1-Am in association their corresponding grounded truth descriptor values Dgt1- Dgtm, supplied as the user data input (cf. figure 1);
- a library 12 of elementary functions EF1, EF2, EF3, ..., which serve as the basic building blocks from which compound functions CF are created on a guided - or constrained — random basis. A selected compound function, or possibly a selected group of compound functions, shall become an outputted DE function 4;
- a heuristics database 14, which contains different types of guiding or constraining rules that come into play in conjunction with random selection events, notably at different stages in the elaboration of compound functions, as shall be explained in more detail below;
- a formal rules and rewriting rule database 15, which contains a set of deterministic rules for recasting automatically-generated compound functions into their formally correct and most rational form;
- a prior results cache 16, which stores results of previously calculated parts of compound functions in view of

obviating the need to recalculate them when subsequently encountered; and

- a validation database 18, which contains the same type of data as the learning database 10, but for other music titles. The audio data contained in that database are not used as reference for elaborating the compound functions, and thus constitute a neutral source for ultimately testing the validity of a candidate DE function 4 selected among the compound functions.

[0054] The signal processing and overall management of the system are carried out by a main processor unit 22 which runs programs contained in a main program memory 24. A user interface 26 associated to a monitor 28, keyboard 30 and mouse 31 allows the user input and output data of figure 1, as well as the internal programming data, to be entered and extracted.

[0055] Figure 3 illustrates the principle of an elementary function EF as exploited by the system 2. Being effectively an operator, the elementary function comprises executable code and one or a set of parameter(s) which it can receive as input Pin, and which defines the elementary function's boundary conditions. An elementary function acts on an operand, or argument 32 — which can be signal data or the output of a preceding elementary function — and generates an output that is the result of the code executed on the operand data. An elementary function EF is catalogued in the system inter alia by the type of operand, designated Toper, on which can operate and on the type of output, designated Tout, it delivers. Types Toper and Tout can be the same or mutually different for a given elementary function. Typical types include: signal, numerical (single number, float, range), vector, or matrix. As explained further, the system 2 treats elementary functions EF — which can be assimilated to modules — as symbolic objects or as executable operators depending on the nature of the processing required in the course of elaborating a compound function CF.

[0056] Figure 4 illustrates an example of an elementary function in the form of a low pass filter (LPF) operator. As such, its executable code comprises a digital LPF algorithm and its input parameters Pip are the cut-off frequency F and optionally the attenuation rate (dB/octave). The operand and output types are respectively Toper=Signal and Tout=Signal.

[0057] Figure 5 illustrates another example of an elementary function, this time in the form of a short-time fast Fourier transform (short-time FFT) operator. The executable code comprises a short time FFT algorithm, and its input parameters Pin are the sampling window and summation limits. The operand and output types are respectively Toper=Signal and Tout=matrix.

[0058] Figure 6 illustrates the principle of a string of elementary functions, the example concerning three elementary functions EFa, EFb and EFc forming a term TCF of a compound function that operates on a signal data S of an audio file, the term being TCF=EFc.EFb.EFa*S. Note that in such a string of elementary functions, an elementary function also constitutes an argument, or operand, for its left-hand neighbour (i.e. succeeding function) to which its is joined by "*" function when the case arises. Also, an output of an elementary function can include parameter input data for its neighbouring function. This is illustrated in figure 6 by the output of function EFb, which produces inter alia a signal which conveys a parameter Pin for its downstream function EFc, for instance the value of a high-pass cut off frequency if the latter is a high-pass filter function.

[0059] A compound function CF can contain an arbitrary number of elementary functions related by different arithmetical operators (+, -, * or ÷). Elementary functions connected together by a multiplicative or divisional operator form a term; several terms can be linked by associative operators + and - as the case arises when constructing a compound function CF.

[0060] Among the programs stored in the main program memory 24 are:

- a compound function construction program 25, which has the role of generating compound functions by assembling together a number of elementary functions EF. The latter are typically signal or data processing functions that can each be considered as a single unit operator or module that produces a determined technical effect on the signal data Si of an audio file or on the output of another elementary function, and
- a function execution program 27, which is composed of the compound functions themselves, these being exploited no longer as symbolic objects, but as executable algorithmic entities for producing technically meaningful operations on signal data S.

[0061] These two programs 25 and 27 are under the overall control of a master program 29 which manages the overall system 2.

[0062] The compound function construction program 25 is based on genetic programming techniques following an artificial intelligence (AI) approach. Accordingly, the elementary functions EF are also handled as symbols, whereby they are treated as first class obj ects in their symbolic representation.

[0063] Thus, the system 2 is capable of handling the elementary functions both as objects, when executing the compound function (CF) construction program 25, and as executable operators, notably for evaluating and testing the compound functions, when executing the function execution program 27. To this end, these two programs 25 and 27 use languages adapted respectively to handling objects and to carrying out numerical calculations, an example of the latter being the "Matlab" language.

[0064] Table I gives a non-exhaustive example of elementary functions stored in the elementary function library 12, together with their operand type Top, output type Tout and parameters.

Table I:

sample list of elementary functions used by the system 2.				
I.1 — Mathematical functions				
Function name	**Operation**	**Param Pin**	**Toper**	**Tout**
DERIV	Time derivative	-	Signal	Signal
MAX	Max value of set	-	set of No.s	No.
MIN	Min value of set	-	set of No.s	No.
SQUARE	Raise power 2	-	No.	No.
LOG	Logarithm	-	No.	No.
MEAN	ave value of set	-	set of Nos.	No.
VAR	variance of set	-	set of Nos.	No.s
ABS(V)	Absolute value |V|	-	signed V	unsigned V
SUM	Summation of terms		No.	set of No
SQRT	Square root	-	No.	No.
POWER	Raise power 'i'	Integer i	No.	No.
I.2 — Signal processing functions				
Function name	**Operation**	**Param Pi**	**Toper**	**Tout**
ENV.	Envelope of signal	-	Signal	Signal
FFT	Fast Fourier transf.	limits	Signal	Signal
stFFT	short-time FFT	limits/time	Signal	Matrix/Vector
AUTOCOR	autocorrelation	-	Signal	Vector
COR	correlation	-	Signal/Signal	Vector
LPF	Low-pass filter	Fcutoff/atten.	Signal	Signal
HPF	High-pass filter	Fcutoff/atten.	Signal	Signal
BPF	Bandpass filter	Flow/Fhigh/atten.	Signal	
Signal				
FLAT	Flatness		Signal	No.
E	Energy		Signal	No.
PITCH	Pitch	-	Signal	No.
1.3- Combining and connecting functions				
Function name	**Operation**		**Para Pi -**	
COMPOSITION o -				
LOOP*	Repeat until	No. iterations		
(	bracket			
COMBINATION *	Multiply		-	-
÷	Divide	-	-	
+	Add	-	-	
-	Subtract	-	-	

* Loop: Output of an iteration can be the input parameter for the next iteration.

[0065] The last four combination operators are simply arithmetic operators which join successive functions, but are treated as functions too.

[0066] Advantageously, when the system handles the elementary functions as symbols, as in the above construction phase, it uses a tree structure.

[0067] According to the tree structure, a compound function CF is symbolised in terms of nodes, where each node corresponds to one elementary function EF, and in which branches connect the nodes according to the arithmetic operators +, -, *, ÷ used.

[0068] As an example, figure 7 illustrates the tree structure for the compound function CF = MAX.DERIV.FFT.FFT. LPF(B1)(S) + ABS.PITCH.LPF(B2)(S) + PITCH.HPF(VARIANCE(S))(S). The three terms are developed along three respective branches Br1-Br3. The three branches join at the "+" function, which is the common link to CF. The order of appearance of the elementary functions is followed along successive nodes, the first elementary function (i.e. the first to operate on the signal) being nearest the free end of its branch.

[0069] The CF construction program 27 initially begins by selecting and aggregating elementary functions in a random fashion.

[0070] Elementary rules and heuristics intervene in this random process to govern the appropriateness of combinations of elementary functions, notably as regards the incorporation of a potential elementary function in the context of any elementary function already present in term under construction.

[0071] Firstly, rules govern the function generation process on a number of different considerations, among which are:

i) **Formal rules.** These rule out the existence of two combined elementary functions EFbEFa if their types are not compatible. In other words, if for the above two functions the output type Tout(a) of EFa is not the same as the operand type Toper(b) of EFb, then EFbEFa, and elementary function EFa has already been selected, then elementary function EFb is attributed a zero weighting coefficient for the random draw that is to select an elementary function for which elementary function EFa is the operand. For example, the formal rule weighting scheme would forbid the meaningless operator combinations FFT.MAX.DERIVABS(V), etc.

The formal rules also ensure that the right-hand most function of a term in the compound function has a signal operand type (Toper=S), given that it will necessarily operate on the signal Si from an audio file.

ii) **Boundary condition rules.** These rules serve to impose constraints on the compound functions or their populations having regard to the system parameters, such as: length constraint on the compound functions, by weighting the number of elementary functions used to favour a prescribed median value, the number of branch points (cf. the tree structure), the number of compound functions produced to form a first population P, etc..

[0072] Secondly, knowledge-based heuristics generally operate by associating to each elementary function EF a weighting coefficient affecting its random draw probability. These coefficients are attributed dynamically according to immediate context. The heuristics can in this way rule out some combinations of elementary functions through a zero weighting coefficient, at one extreme, and force combinations by imposing an absolute maximum value coefficient at the other extreme. A set of intermediate weighting coefficient values is provided to allow the random process to determine the construction of compound functions, albeit with constraints. These heuristics are generally derived from experience in using the system and the user's formal or intuitive knowledge. They thus allow the user to inject his or her know-how into the system and afford a degree of personalisation. They can also be generated by the system itself on an automated basis, using algorithms that detect similarities between compound functions that have been recognised as successful.

[0073] By using the range of attributable weighting coefficients in implementing these heuristics, the system user can use them:

i) as a positive influence, i.e. to encourage the presence or combinations of elementary functions that are of interest. For example, the system uses a knowledge based heuristic to favour the presence of two successive FFTs on a signal S, i.e. FFT.FFT(S), this being found to be conducive to interesting results;

ii) as a negative influence, i.e. that on the contrary to seek to prevent elementary function combinations that are considered to be ineffective or technically inappropriate. For instance, it has been found that the presence of three successive FFTs on a signal S, i.e. FFT.FFT.FFT(S) does not usually produce interesting results. The corresponding heuristic used by the system will thus give a low weighting coefficient to an FFT elementary function in the draw for the elementary function to be the operand on the existing combination of FFT.FFT.

[0074] Before the newly-formed compound functions are processed, they are advantageously submitted to rewriting by application of rewriting rules stored in database 15. Rewriting involves recasting compound functions from their initial form to a mathematically equivalent form that allows them to executed more efficiently. It is governed by a set of deterministic rewriting rules of varying levels of complexity which are executed on each function CFi of the population by the main processor 22, those rules being in machine-readable form.

[0075] Simple rewriting rules eliminate self-cancelling terms in a compound function. For instance, if the compound function considered contains the terms HPF(S, Fa)+FFT(S)- FFT(S), the rewriting rules shall tidy up the expression and reduce it to HPF(S, Fa).

[0076] Another category of rewriting rules eliminates elementary functions that are redundant given their environment, i.e. which do not produce a technical effect. For instance, if an expression contains a bandpass filtering function with a passband between frequencies Fb and Fc, then the rules would eliminate any subsequent function in that term which filter out frequencies outside that passband range, i.e. which are no longer present.

[0077] Other rewriting rules conduct simplifications of a more advanced type. For instance, they will replace systematically the expression E(FFT(S)) by the equivalent, but more easily calculable, expression E(S).

[0078] The implementation of the rewriting rules uses the tree structure of the compound function under consideration. Each node, or section of the tree, is scanned against the set of rewriting rules. Whenever a rewriting rule is applicable to a node or a succession of nodes of the part of the tree being analysed, the node or succession of nodes in question is rewritten according to that rule and replaced by a new tree section or node that corresponds to the thus rewritten — and hence simplified — form of the compound function.

[0079] Each time the tree is modified in this way, it is scanned again, as its new form can create new opportunities for applying rewriting rules that were not evidenced in the previous form of the tree. Accordingly, the tree scanning is repeated cyclically until no changes have been brought for a complete scan.

[0080] To ensure that there is no risk of falling into infinite loops, the rewriting rules do not produce a change that in itself leads to another change, and conversely, ad infinitum. For instance, the system would not contain simultaneously a rule to rewrite A+B as B+A and another rule to rewrite B+A as A+B (in fact, this would be the same rule, infinitely applicable to the result of its own production, and therefore yielding an unending loop)

[0081] Once the population P of compound functions has been formed in accordance with the above heuristics and rules, the compound functions cease to be considered as symbolic objects and are treated instead by the function execution program 27 according to their specified functional definitions.

[0082] Specifically, a compound function CFi (1≤ i ≤ n) is treated by the system 2 as a calculation routine using "Matlab" language and made to operate on the music file data signals Sj (1≤j≤m) stored in the learning database 10 to produce an output value Dij=CFi*(Sj). The signal Sj in question corresponds to a digitised form of an amplitude (signal level) evolving in time t, the time frame of t typically being on the order of 200 seconds in the case of a music title.

[0083] Each of the n compound functions CF1-CFn operates in this way on each of the m titles stored in the learning database 10, thereby producing a total of n.m output values Dij (for i=1 to n and j=1 to m) according to a matrix for the population P. This combination of calculation events is illustrated symbolically in figure 8.

[0084] As shown in figure 8, the n.m output values are mapped in matrix MAT(P) which is stored in a working memory of the main processor 22. These values are accessed at a subsequent stage of evaluating the overall fit of each of the n compound functions CF1-CFn with the descriptor De for which the grounded truths Dgt1-Dgtm were produced. This evaluation is carried out by standard statistical analysis techniques. In the illustrated example, each of the output m.n output values of the matrix MAT(P) is compared with its respective corresponding grounded truth descriptor value Dgti. Specifically, the set of m.n values Dij is analysed against corresponding grounded truth descriptor values Dgt1-Dgtm for the descriptor De ascribed to the respective music titles T1-Tm.

[0085] For a given compound function CFi, the analysis here involves comparing the value Dij with the Dgtj value for the corresponding audio file. This comparison is performed for each of the audio files, so yielding m comparison values. These comparison values are submitted to statistical analysis to obtain a global fit — or fitness — value FIT (afj) with respect to the descriptor De for that function CFi. The global fitness value FIT(afj) expresses objectively how well overall the descriptor values generated by the function CFj match — or correlate — with the corresponding grounded truth descriptors Dgt1-Dgtm.

[0086] The global fit in question is evaluated in the form appropriate for the descriptor, for instance numerical closeness for a numerical descriptor, Boolean correspondence for a Boolean descriptor, etc.

[0087] The above comparisons and statistical analysis are conducted for each of the n compound functions CF1-CFn, and the respective fitness values FIT(af1)-FIT(afn) are stored.

[0088] Then a new population P1 of r compound functions is produced by taking for its members those of the n compound functions CF1-CFn which yield the r best overall fit values (r<n).

[0089] The basic comparisons and analysis in conducting the above procedure is indicated in the algorithm below:

[0090] For CF1: comp. D11 with Dgt1; D12 with Dgt2; D13 with Dgt3; ...; D1m with Dgtm => STATISTICAL ANALYSIS => fit of CF1 with respect to descriptor De = FITaf1(De);

[0091] For CF2: comp. D21 with Dgt1; D22 with Dgt2; D23 with Dgt3; ...; D2m with Dgtm => STATISTICAL ANALYSIS => fit of CF2 with respect to descriptor De
= FITaf2(De)

[0092] For CF3: comp. D31 with Dgt1; D32 with Dgt2; D33 with Dgt3; ...; D3m with Dgtm => STATISTICAL ANALYSIS => fit of CF3 with respect to descriptor De = FITaf3(De) ;

....

[0093] For CFn: comp. Dn1 with Dgt1; Dn2 with Dgt2; Dn3 with Dgt3; ...; Dnm with Dgtm => STATISTICAL ANALYSIS => fit of CF3 with respect to descriptor De = FITafn(De).

→New population P1 = set of r compound functions CF yielding the **r** best fits FITaf(De).

[0094] The r compound functions CF(1)1 to CF(1)r of the new population P1 are then processed in their symbolic object form according to the above-described tree structure. The aim here is to generate from that population P1 a next generation population P2 of compound functions. Advantageously, the system achieves 2 this by using genetic programming techniques. These programming techniques model aspects of biological regeneration or reproduction process naturally ocurring at chromosone level, such as crossover and mutation. In this case, the analogue to a chromosone is an elementary function EF in its symbolic representation.

[0095] Genetic programming is in itself well documented, but hitherto reserved only to fields remote from electronic signal processing. Remarkably, it can be implemented to a great advantage in the present field by virtue of the present approach in which the compound functions question, whose primary purpose is to operate on an electronic signal, are conveniently made exploitable, at critical phases of their elaboration process, as symbolic objects. This "object" form, which advantageosly uses the above-described tree structure, thereby becomes amenable to genetic programming using standard knowledge of applied genetic programming. Accordingly, detailed aspects involving normal knowledge of genetic programming language and practice accessible to a person skilled in the art of genetic programming shall not be detailed in the present description for reasons of conciseness.

[0096] The concept of genetic programming applied to the present signal procesing functions CF is illustrated in connection with two interesting aspects: crossover and mutation. Each is implemented with adapted and specific rules and heuristics stored in the heuristics database 14 and the rules database 15. Among the rules and heuristics applied in the context of genetic programming are the formal and boundary condition rules, and knowledge-based heuristics outlined above, and adapted to circumstances. Overall, the rules and heuristics applied ensure that the compound functions resulting from genetic programming operations are formally acceptable, have a potential for exhibiting an improvement (in terms of fitness) compared the functions from which they are generated, and remain within the system's operating limits.

[0097] **Crossover.** Simply stated, crossover involves taking two compound functions, say CF(1)p and AP(1)q, (for population P1) and creating from them a new function CF(1)pq which contains a mixing of functions CF(1)p and AP (1)q, in a manner analogous to two chromosomes combining to form a new chromosome.

[0098] An example of a new function CF(1)pq produced by crossover of functions CF(1)p and AP(1)q is illustrated by figure 9 using the tree representation. In this representation, the elementary functions are designated in their abbreviated form: ep1-ep10 for compound function CF(1)p and eq1 to eq10 for compound function CF(1)q.

[0099] Crossover is carried out by a crossover generator module 33 forming part of the compound function construction program 25 stored in memory 24. The module 33 receives the two functions CF(1)p and CF(1)q as input and analyses their tree structure using a set of stored crossover rules and heuristics. The analysis seeks to determine, for each function, a suitable break point along a branch. The break point divides the tree in question into a portion that is to be rejected and a portion that is to be retained. In the example, it can be seen that for compound function CF(1)p, the part of the tree structure comprising elementary functions ep7 to ep10 is retained, and the part on the other side of the break point comprising elementary functions ep1 to ep6 is rejected. Similarly for compound function CF(1)q, the part of the tree structure comprising elementary functions eq1 to eq6 is retained, and the part on the other side of the break point comprising elementary functions eq7 to eq10 is rejected. The two retained portions of the respective trees are joined together at their respective break points. This is carried out by attaching with a straight branch the nodes of the respective retained parts lying adjacent the break points. Thus, in the illustrated example, node eq6 is attached by a branch to node ep7. The resultant crossover tree corresponding to compound function CF(1)pq is then composed of elementary functions eq1-eq6, ep7-ep10.

[0100] More complex crossover operations can involve extracting at least one section of a tree (not necessarily an end section) and inserting it within another tree by producing one or several break points in the latter depending on where it is to be accommodated.

[0101] The break points are determined in a guided — or constrained — random draw, in which the guidance is provided by a set of crossover rules and heuristics.

[0102] A first such rule is of the formal type, and requires that two nodes susceptible of being joined together must be formally compatible from the point of view of types, as described above in the context of formal rules. To this end, candidate break points for the random draw are considered in mutually indexed pairs, each member of the pair being associated to a respective tree. The corresponding nodes to be joined are identified in terms of which ones correspond respectively to the operand and the operating function among the pair. Only those pairs of break points satisfying the formal requirements are accepted as candidates.

[0103] Thus, in the illustrated example, the rules in question shall ensure that despite the crossover resulting from a random draw, the operand type Toper(ep7) of elementary function ep7 is the same as the output type Tout(eq6) of elementary function eq6.

[0104] Another rule is of the boundary condition type and requires that the break point should preferably be at the central portion of the tree, e.g. by using weighted random draws, to ensure that the size of crossover-generated compound functions shall be statistically similar in size over repeated generations.

[0105] Finally, knowledge-based heuristics are tested on crossover-generated compound functions. The operators in the new compound function are tested one by one starting from the break point. The knowledge-based heuristics provide a probability for each new operator, regarding which the compound functions is accepted or rejected at each step.

[0106] **Mutation.** Mutation involves taking one compound function CF(1)s and forming a variant thereof CF'(1)s. The variant can be produced by modifying one or a number of the parameters of CF(1)s, and/or by modifying the function's structure, e.g. by adding, removing or changing one or several of its elementary functions, or by any other modification.

[0107] An example of a new compound function CF'(1)s produced by mutation of a function CF(1)s is illustrated by figure 10. In this representation, the initial compound function CF(1)s has a tree structure formed of elementary functions es1 to es7 as shown.

[0108] This function is inputted to a mutation generator module 34 forming part of compound function construction program 25. The mutation generator module 34 produces on that function one or several mutations on a guided - or constrained - random basis.

[0109] In the illustrated example, the outputted mutated function CF'(1)s happens to differ from the inputted function CF(1): i) at the level of the elementary function es6, which is a lo pass filter operator whose parameter P'(es6) now specifies a cut-off frequency of 450 Hz instead of 600 Hz in its original form P (es6), and ii) at level of elementary function es1, which is simply being deleted.

[0110] The mutation process is governed by mutation rules and heuristics, which include formal rules that likewise ensure that any changed function remains formally correct, and boundary condition rules which govern the nature and number of mutations allowed, etc.

[0111] The system can implement other genetic programming operations. For instance, it can produce a cloning, which involves taking one compound function CF(1)t and forming a variant thereof CF'(1)t. The variant has exactly the same functional structure as the original function CF(1)s. Only the values of the fixed parameters are modified. For instance, if the original compound function contains a low-pass filter with a fixed cutoff frequency value of 500Hz, a clone would be the same compound function with a different cutoff frequency value of 400Hz for instance. A cloning parameter can control the extent of the variations of the values (for example +/- 10%). Note that cloning is simply a special — and restricted — case of mutation in the sense described above.

[0112] The genetic programming procedure comprising the above crossover and mutation operations (and possibly other operations) are applied to the population P1 of functions over a given period or number of cycles. When the procedure is terminated for the population, there results a new population P2 of compound functions which are the genetic descendants of those from population P1.

[0113] The number of compound functions CF(2) forming the population P2 is made to be the same as for population P, so as to accommodate for a selection of the r best fitness functions of that population to produce its own succeeding population of functions P3. In order to keep the population size constant, the cumulated proportions of compound function generated randomly (R%), by mutation (M%), by crossover (CO%), and cloning(C%), has to be so that R + M + CO + C = 100%. This consideration applies to all succeeding generations so that their populations do not dwindle in the course of eliminating the lowest fitness functions. Thus, the creation of new population typically calls for a repetition of the random creation procedure (described above for randomly creating the initial population P) to top up the population, given that crossover operations tend reduce the population (if C < CO).

[0114] The new population P2 is then treated in the same manner as the initial population P, starting with a phase undergoing rewriting rules (the rules and heuristics listed above have already applied explicitly or implicitly to that population P2 in the course of the genetic programming (crossover and mutation) operations.

[0115] Accordingly, the correlation, or fitness of each compound function CF(2) of the new population is determined against the grounded truth descriptor values Dgt1 to Dgtm for the descriptor De. The procedure here is just as for obtaining population P1, and the algorithm described above applies mutatis mutandis by replacing P with P1, and P1 with P2.

[0116] The result gives a new set of the r best compound functions CF(2)1 to CF(2)r for the descriptor De, forming the new population P2.

[0117] The above procedure is carried out iteratively over a given number of cycles, each cycle producing a new population Pu from the previous population Pu-1 by genetic programming and a selection of the best compound functions for the population Pu.

[0118] **Implementation of heuristics.**

[0119] Further aspects of the heuristics used by the system are outlined below, notably for function generation (producing the population P) and genetic programming.

[0120] A heuristic can be represented as a function which has for argument (operand):

i) a current term: one or more functions or a tree section, corresponding to the existing environment in terms of the composition of elementary functions EF - for instance the elementary function combinations that have already

been produced during an ongoing function construction process;
ii) a potential term: likewise one or more functions or a tree section, for which the possibility of incorporation into the current term is to be considered by the heuristic.

[0121] The heuristic function produces from the above argument a result in the form of a value in a specified range, e.g. from 0 to 10, which expresses the appropriateness or interest of constructing a function in which the potential term is branched (according to the tree representation) to the current term, e.g. as its argument.

[0122] The range of weighting coefficients (which are here expressed to one decimal) expresses quantitatively the following:

weighting coefficient

0	potential term forbidden from random draw
1	of very little interest
...	
5	of medium interest
9	extremely interesting
10	potential term imposed (i.e. must be selected).

The heuristic function(s) can come into play in the following example: current term = LPF(500Hz).FFT.S
potential term (to become the argument (operand) of the current term) = FFT.DERIV.FFT.S

[0123] A heuristic shall determine the appropriateness of creating the branching where the "S" of the current term becomes "FFT.DERIV.FFT.S".

[0124] In the above case, one example of an applicable heuristic function is the one, which is here designated "HEURISTIC 245", that on the one hand favours the presence of two FFTs (FFT.FFT.(...), and on the other hand discourages the presence of three FFTs (FFT.FFT.FFT.(....). It is catalogued in the heuristics database 14 as:

HEURISTIC245:

- statement of purpose: "interesting to have FFT of FFT, but not FFT of FFT of FFT";
- form: HEURISTIC245(current term, potential term);
- potential term weighting coefficient attribution procedure:

  if type of current term is FFT,
  AND if current term does not contain other FFT type terms,
  AND if type of potential term is FFT,
  AND if potential term contains an FFT,
  THEN : potential term's weighting coefficient = 0.1 {indeed, the complete function would then have three FFTs, and a low weighting coefficient is therefore attributed}
  ELSE: potential term's weighting coefficient = 8.0.
  Procedures and statements of which the above is an example can be adapted to all other heuristics of the database 14.

Another heuristic function designated HEURISTIC250 is as follows:

HEURISTIC250:

- statement of purpose: "give preference to a filtering on raw signals".
- potential term applicable: Filter class {LPF, HPF, BPF..}
- form HEURISTIC250(current term, filter class)
- potential term weighting coefficient attribution procedure:

  if current term contains FFT, THEN: potential term's weighting coefficient = 0 {filtering is meaningless if an FFT is carried out beforehand},
  if current term contains CORRELATION, THEN: potential term's weighting coefficient = 3 {if a correlation is carried out beforehand, filtering is of doubtful use, but could nevertheless return an interesting value},
  ELSE: potential term's weighting coefficient =7 {if the current term does not contain signal modification

operations such as FFT, CORRELATION, it is generally useful to filter the signal to retain just some of its spectral components}.

Other heuristics can be implemented to take in account a given context, or an indication of the descriptor De for which the compound function is constructed. These are referred to as "context sensitive heuristics".

An example of a context sensitive heuristic is as follows:

Context sensitive heuristic CSHEURISTIC280

- statement of purpose: "to treat problems pertaining to a sung voice (presence, extraction, ...), whereby it is useful to use frequencies of the human voice e.g. from 200 Hz to 1500 Hz";
- context = analysis of voice
- potential term to which it is applicable: Filter(lowF, highF)
- current term to which it is applicable: any.
- potential term's weighting coefficient attribution procedure:
    - if lowF (of signal) is close to 200 HZ, potential term's weighting coefficient is correspondingly high (e.g. 9 for 200 Hz, 8 for 300 Hz, etc.);
    - if highF (of signal) is close to 1500, potential term's weighting coefficient is correspondingly high (e.g. 9 for 1500 Hz, 8 for 1400 Hz, etc.).

[0125] A further class of heuristics, known as "reference base sensitive heuristics" takes into account the global nature of the signals in the learning database 10. The latter is expressed by a quantity referred to as "global reference indicator"

[0126] These heuristics therefore additionally have this global reference indicator as their parameter. The latter can also be for instance a set of descriptors taken out from that reference database.

[0127] They enable to select functions in dependence of the nature of the reference signals.

[0128] An example a of reference base sensitive heuristic is as follows:

HEURISTIC465;

- form HEURISTIC465 (current term, potential term, global reference indicator):

- statement of purpose: "indicate that it is particularly useful to use FFTs when the reference database signals overall have a complex spectrum".
- potential term's weighting coefficient attribution procedure:
    - if current term does not contain other FFT type terms,
    - AND if potential term is an FFT,
    - AND if the reference database signals have (for the most part) a complex spectrum, with spectral characteristics SC1, SC2, ..

[0129] THEN: potential term's weighting coefficient = 9.

Caching technique.

[0130] The iterative loops used by the system 2 involve a considerable amount of processing, especially for the steps of extracting a value Dij of a compound function CFi for a signal data Sj. In order to maximise the efficiency of that task, the system advantageously uses the prior results cache 16 as a source of precalculated results that save having to repeat calculations that have previously been performed.

[0131] The corresponding caching technique involves analysing a compound function under execution in terms of its tree structure, and thus involves both the symbolic, object representation of the function and its exploitation as an operator.

[0132] Figure 11 is an example illustrating how the caching technique is implemented. At a time t1, the system 2 is required to calculate the expression MAX*FFT*LPFILTER(F=600Hz)*(Si) (F=cut-off frequency) that appears at a branch Brp of a given compound function CFu(Si).

[0133] Assuming that the prior results cache 24 is initially empty at that stage, the main processor 22 proceeds in a stepwise manner on the successive elementary functions. Thus, it calculates LPF(S), F=600Hz at a first step i) and stores the result as R1, then calculates FFT*R1 at a second step ii) and stores the result as R2, and finally calculates MAX*R2, which yields the value for the term of branch Br1.

[0134] The above intermediate and final values R1, R2 and R3 are sent to prior results cache 24 together with an indication of the parts of branch Br1 that generated them. Thus, the cache records that LPF(Si), F=600Hz = R1, FFT*LPFILTER(F=600Hz)*(Si) = R2, and MAX*FFT*LPFILTER(F=600Hz)*(Si) = R3 in a two-way correspondence table. Note that results are stored in the cache 24 for an operation on a specific set of data contained in the signal data Si. The set in question can correspond to a predetermined time sequence of the associated audio file, for instance corresponding to one sampling event.

[0135] At a later time t2, the main processor 22 is required to calculate the value of a branch Brq belonging to another function CFv(S). In the example, the branch Brq corresponds to the term AVE* FFT*LPFILTER(F=600Hz)*(Si).

[0136] The cache 24 now no longer being empty, the main processor 22 proceeds to determine first whether at least one function of that branch has already been calculated and stored in the cache 24. To this end, it performs a scan routine on branch Brq by determining whether the first function to be calculated, i.e. LPFILTER(F=600Hz)*(Si) is indexed in the cache 24. The answer being yes, it determines whether the first and second functions together, i.e. FFT*LPFILTER (F=600Hz)*(Si) are indexed in the cache. The answer being again yes, it determines whether the first, second and third functions together, i.e. AVE*FFT*LPFILTER(F=600Hz)*(Si) are indexed in the cache. The answer this time being no, it is thereby informed that the most useful result in the cache is R2= FFT*LPFILTER(F=600Hz)*(Si). Accordingly the main processor 22 rewrites the contents of branch Brj as AVE(R2) and calculates that value. The result of that calculation R4, indexed to the function AVE(R2), or equivalently to the term AVE* FFT*LPFILTER(F=600Hz)*(Si), is sent to the cache 24 so that it need not be calculated in whole at a later stage.

[0137] The cache 24 is thus enriched with new results every time a new function or term is encountered and calculated. The caching technique becomes increasingly useful cache contents grow in size, and contributes remarkably to the execution speed of the system 2.

[0138] In practice, the number of entries in the prior results cache 24 can become too large for an efficient use of allowable memory space and search. There is therefore provided a monitoring algorithm which regularly checks the usefulness of each result stored in the cache 24 according to a determined criterion and deletes those found not to useful. In the example, the criterion for keeping a result Ri in the in the cache 24 is a function which takes into account: i) the calculation time to produce Ri, ii) the frequency of use of Ri, and iii) the size (in bytes) of Ri. The last condition can be disregarded if available memory space is not an issue, or if it is managed separately by the computer.

[0139] After a given number of cycles or a given execution time according to a chose criterion, the system 2 produces as its user data output a descriptor extraction (DE) function 4 (cf. figure 1). The latter is the member of the latest generation population Pf of compound functions CF(f) that has been found to have the best fit for the descriptor De. The user output can produce more than one member of that population, for instance the b best fit functions CF(f), where b is an arbitrary integer, or those compound functions that exhibit a fit better than a given threshold.

[0140] The criterion for ending the loop back to creating a new population of functions is arbitrary, an ending criterion being for example one or a combination of: i) execution time, ii) quality of results in terms of the functions' fitness, iii) number of generations of functions (loops executed), etc.

[0141] Preferably, before an composite function is finally outputted as a DE function for future exploitation, it is validated against signals of other music titles taken from the validation database 18. As these signals are not used to influence the construction of the DE functions 4, they serve as a neutral reference on which to check their effectiveness. The checking procedure involves determining the degree of fit between on the one hand a descriptor value obtained by making a DE function operate on a signal Sv of the validation database and on the other the grounded truth descriptor value associated to the music title of that signal Sv. An overall correlation or validation value is generated by statistical analysis over a given number of entries of the validation database 18. If the validation value is above an acceptable threshold, the DE function 4 is validated and thus considered to be exploitable. In the opposite case, the DE function is rejected and another DE function is considered.

[0142] Figure 12 is a flowchart illustrating some steps performed by the system 2 of figure 2 in the course of producing a descriptor extraction function DE 4, these being:

- inputting user input data to constitute the learning database 10 (step S2), whereby the database comprises the set of reference signals S1-Sm in association with their global characteristic values Dgt1-Dgtm pre-attributed,
- preparing a population P1 of functions CF1-CFr each composed of at least one elementary function (EF) (step S4),
- modifying functions of the current population using programmed means 22, 25 that handle their elementary functions as symbolic objects (step S6),
- operating each function of the population on at least one reference signal using means 22, 27 that exploit the elementary functions as executable operators, to obtain a calculated value for each compound function of the population in respect of the reference signal (step S8),

- for each function of the population, determining the degree of matching between its calculated value and the pre-attributed value Dgti for the signal from which that value has been calculated (step S10),
- selecting functions of the population producing the best matches to form a new population P2 of functions (step S12),
- if an ending criterion is not satisfied, returning to step S6, where the new population becomes the current population (step S 14), and
- if an ending criterion is satisfied, outputting at least one function of the current new population as a general function (4) of the user output (step S16).

[0143] Heuristics and/or rules can be entered, edited, modified through the user interface unit 26 e.g. by manual input (keyboard) or by download, thereby making the system fully adaptive and configurable.

[0144] Typically, the system generates several hundred compound functions over a twelve-hour period. The learning database preferable comprises at least several hundred titles, and preferably several thousand. The handling of such large databases is simplified by the use of the above caching technique and heuristics. Parallel processing, where a same function is calculated on several titles simultaneously using respective processors over a network can also be envisaged.

[0145] The size of the compound functions is typically of the order of ten elementary functions.

[0146] The system is remarkable in that it does not need to be informed of the descriptor De for which it must a find a suitable DE function. In other words, all that is necessary is to provide examples of just the descriptor values Dgti associated to music titles Ti and their signal data Si. This makes the system 2 completely open as regards descriptors, and amenable to generating suitable DE functions for different descriptors without requiring any initial formal training or programming specific to a given descriptor.

[0147] In the embodiment, the system is connected to a network, such as Internet or a LAN, in order to facilitate the acquisition of music titles through a download centre 36. The networking also makes it possible to share and exchange elementary functions, compound functions, heuristics, rules and DE functions found to be interesting, as well as results data for the prior results cache 24, allowing parallel processing, etc. In this way, an interactive community of searchers can be fostered and allow the a rapid spread of new developments.

[0148] The heuristics and/or rules can be entered / edited / parameterised through the user interface unit 26; they can also be generated / adapted internally by the system, e.g. by processing techniques based on analysing compound functions that produce the best fits and determining common features thereof expressible as rules and/or heuristics.

[0149] Figure 12 is an example of different DE functions and their fitness produced automatically by the system for evaluating the presence of voice in music title.

[0150] Figure 13 is an example of different compositions of DE functions in terms of elementary functions, and their fitness produced automatically by the system to evaluate the global energy of music titles.

[0151] The method and data implemented by the system can be presented as executable code forming a software product stored on a computer-readable recording medium, e.g. a CD-ROM or downloadable from a source, the code executing all or part of operations presented.

[0152] From the foregoing, it will be appreciated that the above-described system is remarkable by virtue of many characteristics, inter alia :

- its genericity: the system is independent of a given descriptor, and is able to infer an extractor (DE function) for arbitrary problems;
- its heuristics: the system contains many built-in heuristics that guide the search, and reduce the search space. The originality here is that the system encodes heuristics specific to signal processing, and provides a way to evaluate the fitness of a given function by testing it against a real database of music titles;
- caching, which greatly reduces the workload on the main processor 22 and accelerates calculation considerably;
- rewriting, which provides the groundwork for ensuring that functions shall be calculated in their most rational form;
- implementation: the aim is calculate functions on an automated basis, rather than manually. In the respect, the embodiment can be likened to an expert system in artificial intelligence, where it substitutes the role of the human specialist in signal processing. Extracting descriptors automatically from the digital representation of an acoustic signal in accordance with the invention allows to scale-up descriptor acquisition, and also ensures that the descriptors obtained are objective.

[0153] The remarkable aspects of the present automated system 2 can be appreciated from considering how the task would have to be considered in a manual approach. The starting point is the raw data signals as seen by the specialist in signal processing. The latter tries out various processing functions according to a empirical methodology in the expectation that some rule shall emerge for correlating complex signal characteristics with that descriptor. In other words, the approach is extremely heuristic in nature. It is also largely based on trial and error.

[0154] This task of manually finding a combination of signal processing functions by signal processing experts is time-consuming and subject to many subjective biases, errors, etc. In most cases it would be too impractical to be considered in a real-life application.

System applications.

[0155] 1. Fully autonomous automatic descriptor extraction function generating system.

[0156] In the embodiment described above, the programmed system 2 is able to generate an exploitable DE function 4 from scratch using just the user data input indicated with reference to figure 1.

[0157] The DE function typically takes on the form of executable code or instructions comprehensible to a human or machine. The contents of the DE function thereby allow processing on the audio data signal of any given music title to extract its descriptor De, the latter being referenced to the function.

[0158] The process of extracting in this way the descriptor De of a music title can be performed by an apparatus which is separate from the system. The apparatus in question takes for input the DE function (or set of DE functions) produced by the system 2 and audio files containing signals for which a descriptor has to be generated. The output is then the descriptor value Dx of the descriptor De for the or each corresponding music title Tx. The DE function (or set of DE functions) produced by the system 2 is in this case considered as a product in its own right for distribution either through a network, or through a recordable medium (CD, memory card, etc.) in which it is stored.

2. Descriptor extraction

[0159] It will be noted that the system 2 already includes all the hardware and software necessary to constitute an automated descriptor generating apparatus as defined in the preceding section. In this case, the DE functions shown as user data output of figure 1 are fed back to the system (or kept within system and stored). The system can be switched to the descriptor extraction mode in which audio signal data corresponding to a music file Tx to be analysed is supplied as an input and the corresponding music descriptor value of Tx for the descriptor De is provided as the output.

3. Authoring tool for producing descriptor extraction functions.

[0160] In a variant, the system is implemented more as an authoring tool. In this implementation, the system allows the outputted DE functions to be modified by external intervention, generally by a human operator. The rationale here is that in some cases, while the functions produced automatically may not be strictly optimal, they are nevertheless highly interesting as a starting basis for optimisation, or "tweaking". The advantage in this case resides in that the human specialist has at his disposal a descriptor extraction function firstly which is already proven to be effective compared to a large number of other possible functions, indicating that it possesses a sound structure, and secondly which is proven to be amenable to fast and consistent execution. Note that the DE function outputted by the system 2 can generally be modified by intervening in this case too either at the level of the basic elementary function taken as a symbolic object, e.g. by substitution, removal, or addition, or at the level of the internal parameterisation of a basic elementary function, e.g. by changing a cut-off frequency value in the case of the low-pass filtering elementary function.

4. Evaluation tool for externally produced descriptor extraction functions.

[0161] The aspect of the system 2 that analyses and evaluates compound functions can be put at the disposal of external sources of candidate DE functions, so as to help designers evaluluate their own descriptor extraction functions. The evaluation can be used to provide an objective assessment of the "fitness" FIT of such a candidate function with respect to the learning database 10 or validation database 18.

5. Function calculation tool for externally produced DE functions.

[0162] Similarly, the function calculation potential of the system 2, enhanced notably by the above-described rewriting rules and the caching technique, can be put at the disposal of outside users. The latter can then input a given complex signal processing function (not necessarily in the context of descriptor extraction) and receive a calculated value as an output.

Scope

[0163] While the invention has been described in the context of a system adapted to process audio file signal data to produce descriptor extraction functions DE, it will be apparent that the teachings of the invention are applicable to many other applications where it is required to analyse low level characteristics of an electronic data signal (digital or analogue) in view of extracting higher-level information relating to its contents. For instance, the invention can be implemented for obtaining descriptor extraction functions operative on video or image signal data, the descriptors in this case being applicable to visual contents, such as indicating whether a scene is set at night or daytime, the amount of action, etc. Other applications are in the fields of automatic cataloguing of sound, scenes, objects, animals, plants, etc. through high level descriptors.

Claims

1. Method of generating a general extraction function (4) which can operate on an input signal (Sx) to extract therefrom a predetermined global characteristic value (DVex) expressing a feature of the information (De) conveyed by that signal,

characterised in that it comprises the steps of:

- generating automatically compound functions (CF1- CFn), each compound function being composed of at least one of a set of elementary functions (EF1, EF2, ..), by using means (22, 25) that handle said elementary functions as symbolic objects,
- operating said compound functions on at least one reference signal (S1-Sm) having a pre-attributed global characteristic value (Dgt1-Dgtm) serving for evaluation, by using means (22, 27) that process said elementary functions as executable operators,
- determining the correlation between the values (Dij) extracted by those compound functions as a result of operating on said reference signal and the pre-attributed global characteristic value (Dgt1-Dgtm) of said reference signal, and
- selecting said general extraction function (4) among those compound functions for which said correlation is relatively high.

2. Method according to claim 1, wherein said compound functions are generated in successive populations (P, P1, P2),

wherein each new population of functions takes as a basis earlier population functions which produce a relatively high correlation.

3. Method according to claim 1 or 2, wherein it is performed by the steps of:

a) preparing at least one reference signal (S1-Sm) for which said predetermined global characteristic value (Dgt1-Dgtm) is pre-attributed,
b) preparing a population (P1) of compound functions (CF1-CFr) each composed of at least one elementary function (EF),
c) modifying compound functions of the current population using said means (22, 25) that handle their elementary functions as symbolic objects,
d) operating said compound functions of said population on at least one said reference signal using said means (22, 27) that exploit said elementary functions as executable operators, to obtain a calculated value for each compound function of the population in respect of said reference signal,
e) for at least some compound functions of the population, determining the degree of matching between its calculated value and the pre-attributed value (Dgti) for the signal from which that value has been calculated,
f) selecting compound functions of said population producing the best matches to form a new population (P2) of functions,
g) if an ending criterion is not satisfied, returning to step c), where said new population becomes the current population,
h) if an ending criterion is satisfied, outputting at least one compound function of the current new population as a said general function (4).

4. Method according to any one of claims 1 to 3, wherein said compound functions are produced by random choices guided by rules and/or heuristics.

5. Method according to claim 4, wherein said rules and/or heuristics comprise at least one rule which forbids, from a random draw for selecting an elementary function to be associated with a part of a compound function under construction, an elementary function that would be formally inappropriate for that part.

6. Method according to claim 4 or 5, wherein said rules and/or heuristics comprise at least one heuristic which favours, in a random draw for selecting an elementary function to be associated with a part of a compound function under construction, an elementary function which is considered to produce potentially useful technical effects in association with that part, and/or which discourages from said random draw an elementary function considered to produce technical effects of little or no use in association with that part.

7. Method according to any one of claims 4 to 6, wherein said rules and/or heuristics comprise at least one heuristic which ensures that a said compound function (CF) comprises only elementary functions (EF) that each produce a meaningful technical effect in their context.

8. Method according to any one of claims 4 to 7, wherein said rules and/or heuristics comprise at least one heuristic which takes into account at least one overall characteristic of said reference signals.

9. Method according to any one of claim 2 to 8, wherein a new population (P1, P2, ..) of functions is produced using genetic programming techniques.

10. Method according to claim 9, wherein said genetic programming techniques comprise at least one of following:

- crossover,
- mutation,
- cloning.

11. Method according to claim 10, wherein a crossover operation and/or a mutation operation is guided by at least one heuristic of any one of claims 4 to 8.

12. Method according to any one of claims 1 to 10, wherein said means (22, 25) that handle said elementary functions as symbolic objects manage said functions (CF) in accordance with a tree structure comprising nodes and connecting branches, in which each node corresponds to a symbolic representation of a constituent unit function (EF), said tree having a topography in accordance with the structure of said function.

13. Method according to any one of claims 1 to 12, further comprising a step of submitting a compound function (CF) to at least one rewriting rule executed by processing means (15, 22) to ensure that said compound function is cast in its most rational form or most efficient form in respect of execution efficiency.

14. Method according to any one of claims 1 to 13, wherein it uses a caching technique for evaluating a function, in which results (R1, R2, ...) of previously calculated parts of functions are stored (24) in correspondence with those parts, and a function currently under calculation is initially analysed to determine whether at least a part of said function can be replaced by a corresponding stored result, said part being replaced by its corresponding result if such is the case.

15. Method according to claim 14, comprising the steps of checking the usefulness of results stored (24) according to a determined criterion, and of erasing those found not to be useful, said criterion for keeping a result Ri being a function which takes into account: i) the calculation time to produce Ri, ii) the frequency of use of Ri and, optionally, iii) the size (in bytes) of Ri.

16. Method according to any one of claims 1 to 15, wherein said elementary functions (EF) comprise signal processing operators and mathematical operators.

17. Method according to any one of claims 1 to 16, further comprising a step of validating a general function (CF) against at least one reference signal having a known value for said general characteristic, and which was not used to serve as said reference.

18. Method according to any one of claims 1 to 17, wherein said signal (S) expresses an audio content, and said global characteristic is a descriptor (De) of the audio content.

19. Method according to claim 18, wherein said audio content is in the form of an audio file, said signal (S) being the signal data of said file.

20. Method according to claim 18 or 19, wherein said descriptor comprises at least one among:

- a global energy indication,
- a sung or instrumental audio content,
- an evaluation of the danceability,
- an acoustic or electric sounding audio content,
- presence or absence of a solo instrument, e.g. guitar or saxophone solo.

21. Method of extracting a global characteristic value (DVex) expressing a feature of the information (De) conveyed by a signal (Sx), **characterised in that** it comprises calculating for said signal (Sx) the value of a general function (4) produced specifically by the method of any one of claims 1 to 20 for that global characteristic.

22. Apparatus (2) for generating a general function (4) which can operate on an input signal (Sx) to extract therefrom a predetermined global characteristic value (DVex) expressing a feature of the information (De) conveyed by that signal,

characterised in that it comprises:

- means (22, 25) for generating automatically compound functions (CF1-CFn), each compound function being composed of at least one of a set of elementary functions (EF1, EF2, ..), said means (22, 25) handling said elementary functions as symbolic objects,
- means (22, 27) for operating said compound functions on at least one reference signal (S1-Sm) having a pre-attributed global characteristic value (Dgt1-Dgtm) serving for evaluation, said means (22, 27) processing said elementary functions as executable operators,
- means (22) for determining the correlation between the values (Dij) extracted by those compound functions as a result of operating on said reference signal and the pre-attributed global characteristic value (Dgt1-Dgtm) of said reference signal, and
- means (22) for selecting said general extraction function (4) among those compound functions for which said correlation is relatively high.

23. Apparatus according to claim 22, configured to execute the method according to any one of claims 1 to 21.

24. Use of the apparatus according to claim 22 or 23 as a fully autonomous automatic descriptor extraction function generating system.

25. Use of the apparatus according to claim 22 or 23 as a descriptor extraction means.

26. Use of the apparatus according claim 22 or 23 as an authoring tool for producing descriptor extraction functions (4).

27. Use of the apparatus according to claim 22 or 23 as an evaluation tool for externally produced descriptor extraction functions.

28. A general function (4) in a form exploitable by an electronic machine, produced specifically by the apparatus according to claim 22 or 23.

29. A software product containing executable code which, when loaded in a data processing apparatus, enables the latter to perform the method of any one of claims 1 to 21.

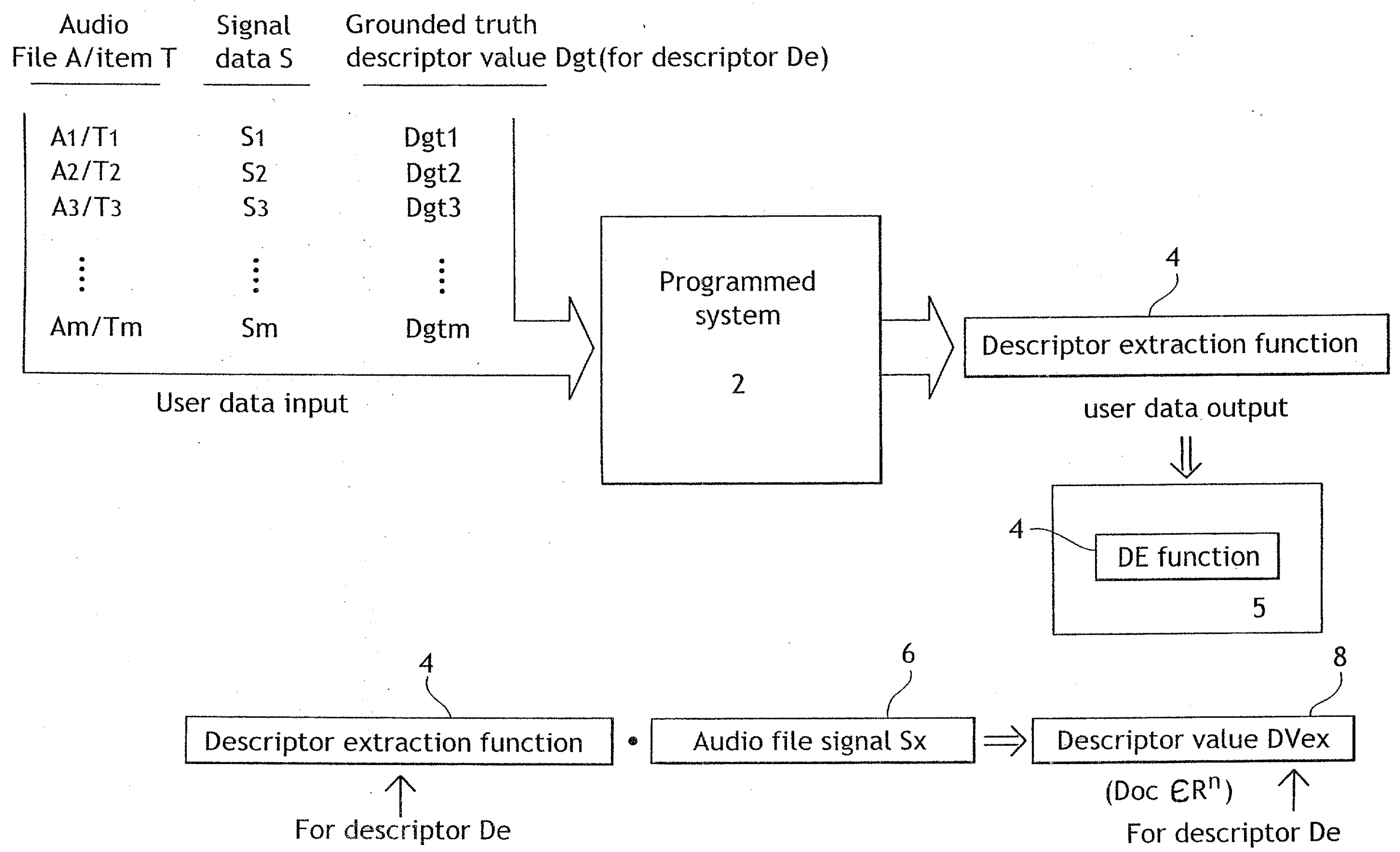

Audio
File A/item T
Signal
data S
Grounded truth
descriptor value Dgt(for descriptor De)
A1/T1
A2/T2
A3/T3
Am/Tm
S1
S2
S3
Sm
Dgt1
Dgt2
Dgt3
Dgtm
User data input
Programmed
system
2
4
Descriptor extraction function
user data output
4
DE function
5
4
Descriptor extraction function
For descriptor De
6
Audio file signal Sx
8
Descriptor value DVex
(Doc ЄR^n)
For descriptor De

FIG.1

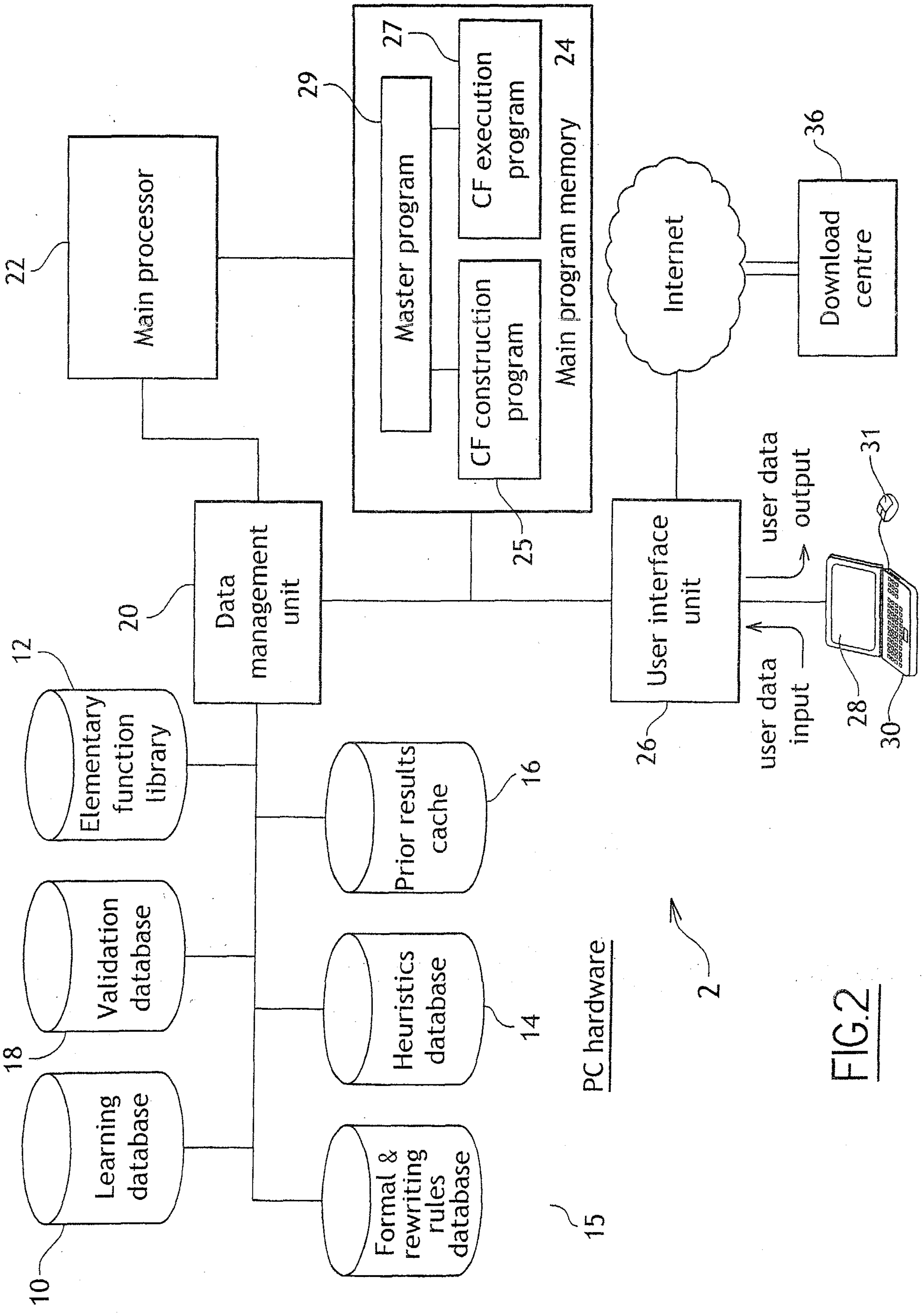

Main processor
22
Master program
29
CF execution program
27
CF construction program
25
Main program memory
24
Internet
Download centre
36
Data management unit
20
User interface unit
26
user data output
user data input
28
30
31
Elementary function library
12
Validation database
18
Learning database
10
Prior results cache
16
Heuristics database
14
Formal & rewriting rules database
15
PC hardware
2

FIG.2

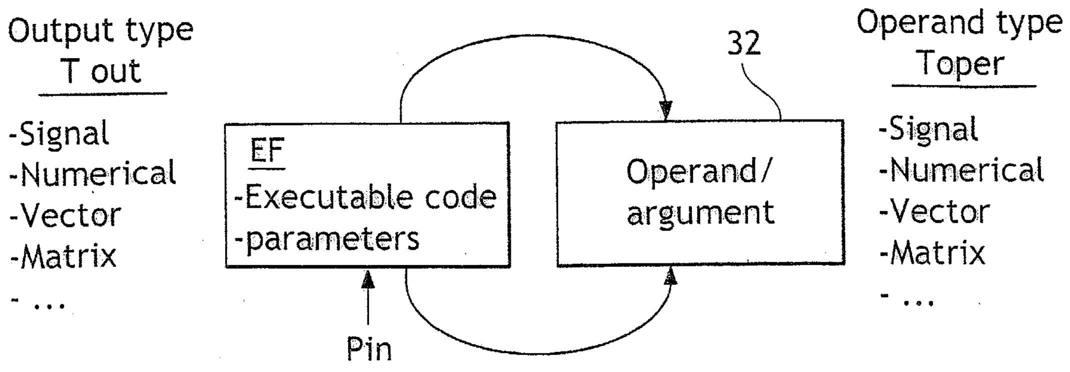


FIG.3

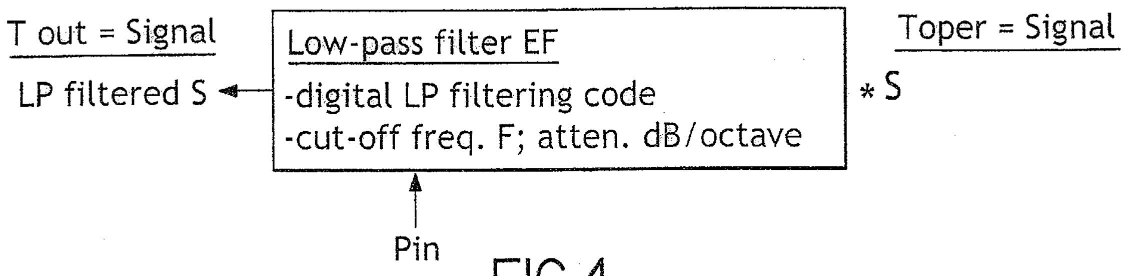


FIG.4

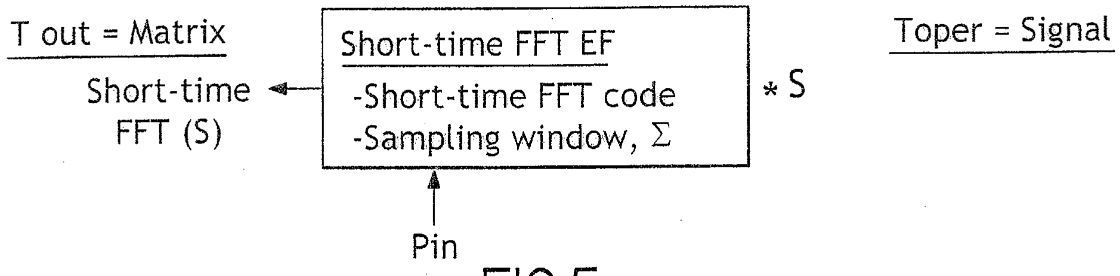
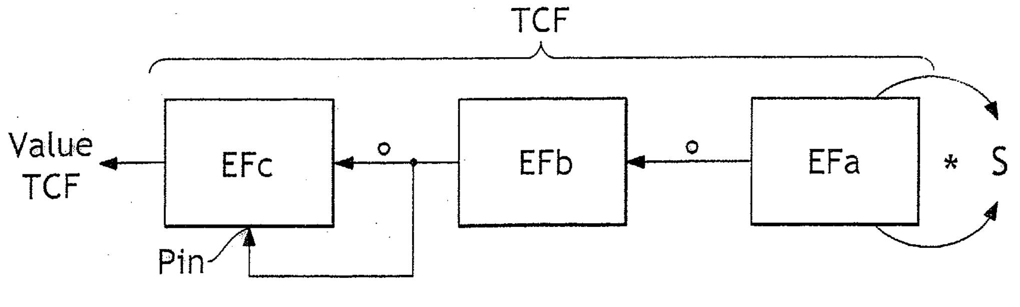


FIG.5



TCF = EFc • EFb • EFa * S

FIG.6

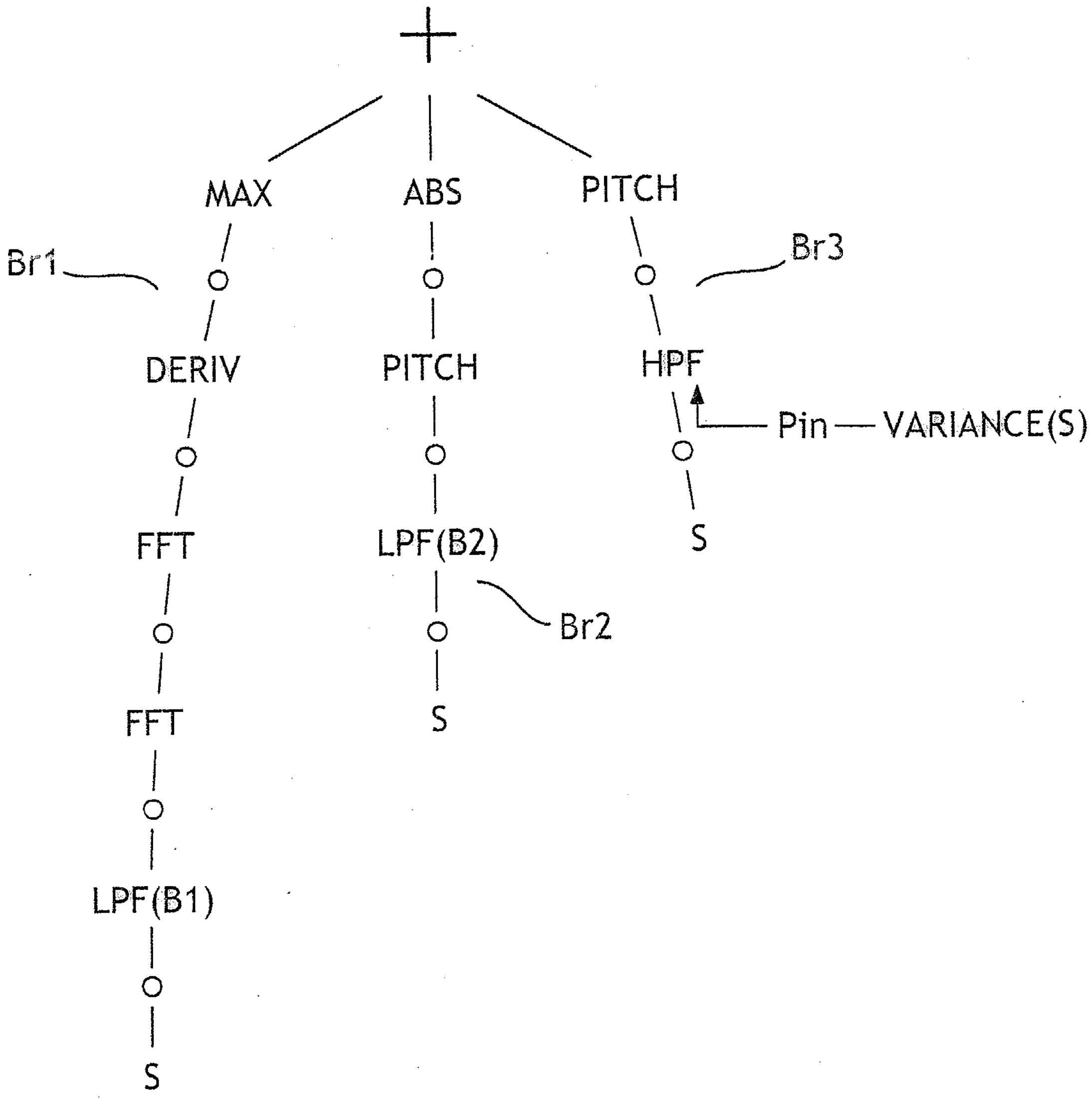

+
MAX
ABS
PITCH
Br1
Br3
DERIV
PITCH
HPF
Pin
VARIANCE(S)
FFT
LPF(B2)
S
Br2
FFT
S
LPF(B1)
S

FIG.7

$$\begin{pmatrix} CF1 \\ CF2 \\ CF3 \\ \vdots \\ CFm \end{pmatrix} \times \begin{pmatrix} S1 \\ S2 \\ S3 \\ \vdots \\ Sm \end{pmatrix} \Longrightarrow \begin{pmatrix} D11 & D21 & D31 \ldots Dn1 \\ D12 & D22 & \boxed{D32} \ldots Dn2 \\ D13 & D23 & D33 \ldots Dn3 \\ \vdots & & \\ D1m & D2m & D3m \ldots Dnm \end{pmatrix}$$

D32 = AF3*(S2) etc.

MAT (P)= matrix of n.m Dij o/p values for population P

For CF1: comp. D11 with Dgt1; D12 with Dgt2; D13 with Dgt3; ...; D1m with Dgtm ⇒ STATISTICAL ANALYSIS ⇒ fit of CF1 with respect to descriptor De = FITaf1(De);

For CF2: comp. D21 with Dgt1; D22 with Dgt2; D23 with Dgt3; ...; D2m with Dgtm ⇒ STATISTICAL ANALYSIS ⇒ fit of CF2 with respect to descriptor De = FITaf2(De)

For CF3: comp. D31 with Dgt1; D32 with Dgt2; D33 with Dgt3; ...; D3m with Dgtm ⇒ STATISTICAL ANALYSIS ⇒ fit of CF3 with respect to descriptor De = FITaf3(De);

....

For CFn: comp. Dn1 with Dgt1; Dn2 with Dgt2; Dn3 with Dgt3; ...; Dnm with Dgtm ⇒ STATISTICAL ANALYSIS ⇒ fit of CF3 with respect to descriptor De = FITafn(De).

→ New population P1 = set of r compound functions CF yielding the r best fits FITaf(De).

FIG.8

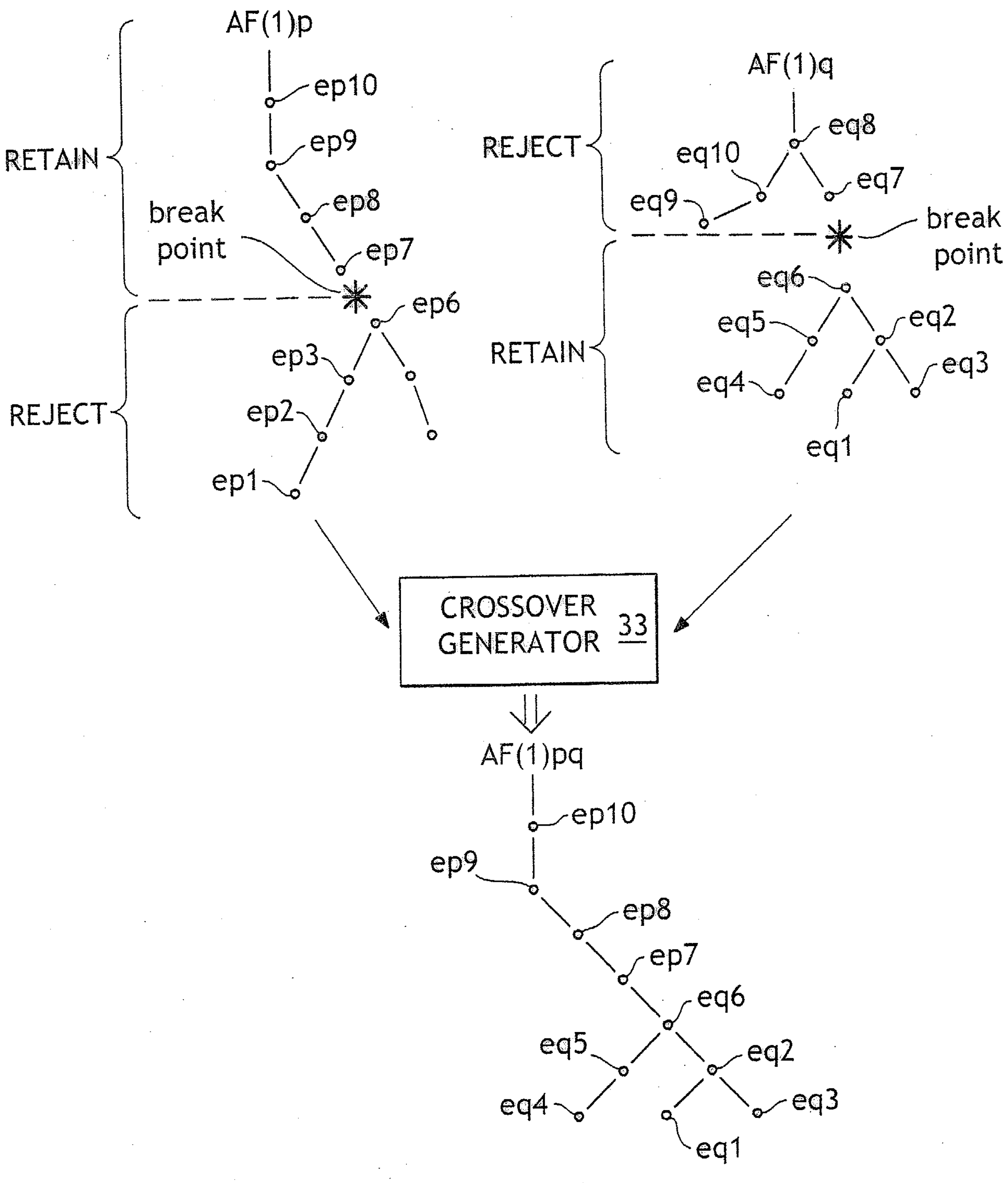

AF(1)p
ep10
ep9
ep8
ep7
RETAIN
break point
ep6
ep3
ep2
ep1
REJECT
AF(1)q
eq8
eq10
eq7
eq9
REJECT
break point
eq6
eq5
eq2
eq4
eq3
eq1
RETAIN
CROSSOVER GENERATOR 33
AF(1)pq
ep10
ep9
ep8
ep7
eq6
eq5
eq2
eq4
eq3
eq1

FIG.9

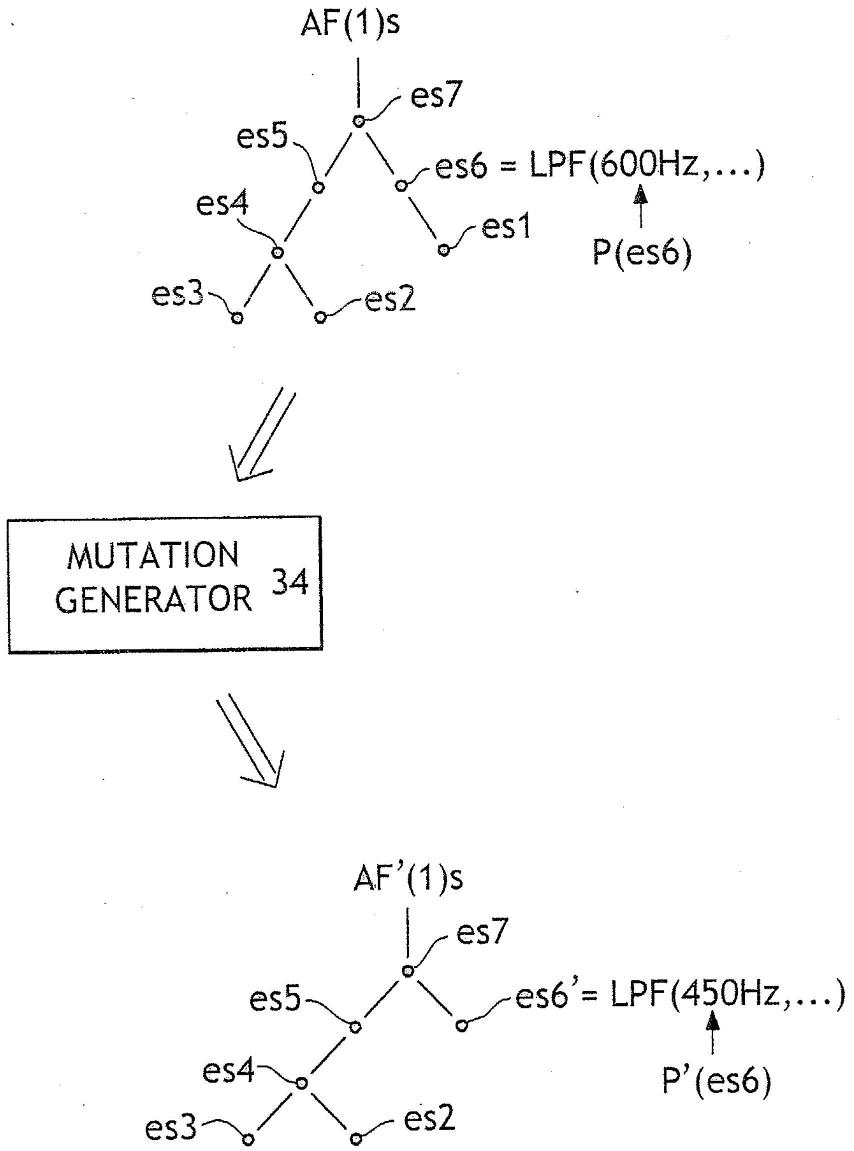

AF(1)s
es7
es5
es6 = LPF(600Hz,...)
es4
es1
P(es6)
es3
es2
MUTATION GENERATOR
34
AF'(1)s
es7
es5
es6'= LPF(450Hz,...)
es4
P'(es6)
es3
es2

FIG.10

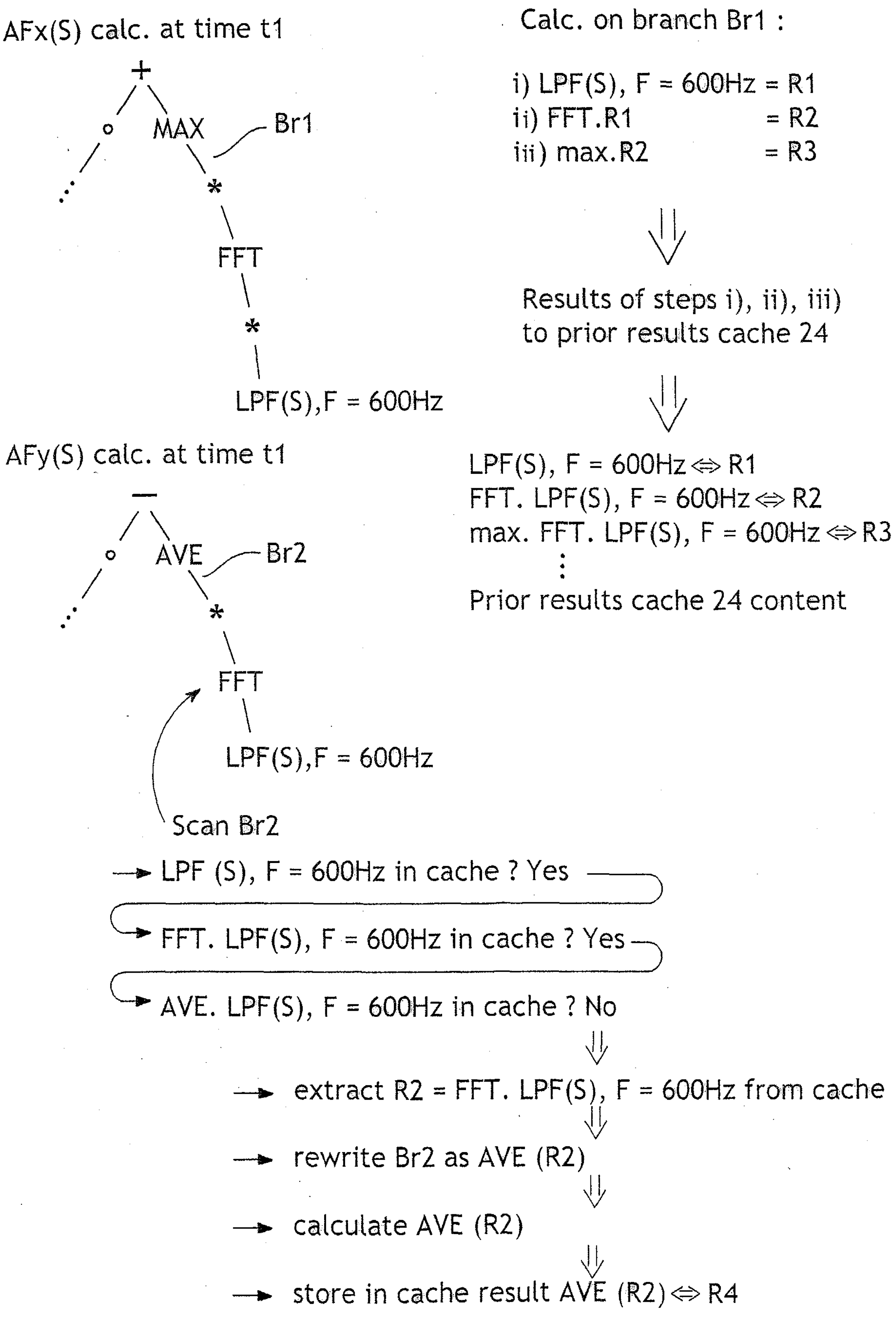

AFx(S) calc. at time t1
+
o
MAX
Br1
*
FFT
*
LPF(S),F = 600Hz
Calc. on branch Br1 :
i) LPF(S), F = 600Hz = R1
ii) FFT.R1 = R2
iii) max.R2 = R3
Results of steps i), ii), iii)
to prior results cache 24
LPF(S), F = 600Hz ⇔ R1
FFT. LPF(S), F = 600Hz ⇔ R2
max. FFT. LPF(S), F = 600Hz ⇔ R3
Prior results cache 24 content
AFy(S) calc. at time t1
—
o
AVE
Br2
*
FFT
LPF(S),F = 600Hz
Scan Br2
LPF (S), F = 600Hz in cache ? Yes
FFT. LPF(S), F = 600Hz in cache ? Yes
AVE. LPF(S), F = 600Hz in cache ? No
extract R2 = FFT. LPF(S), F = 600Hz from cache
rewrite Br2 as AVE (R2)
calculate AVE (R2)
store in cache result AVE (R2) ⇔ R4

FIG.11

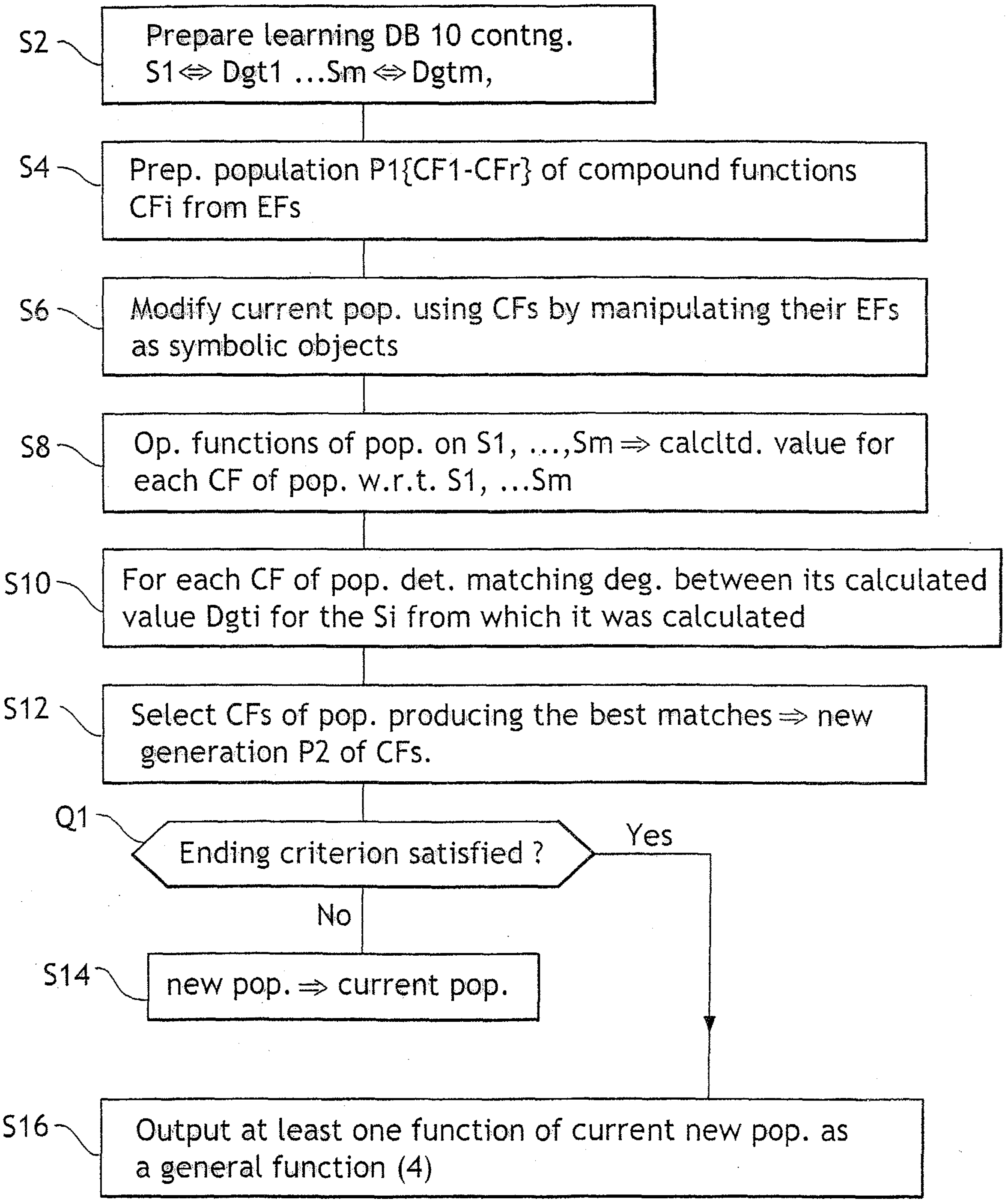
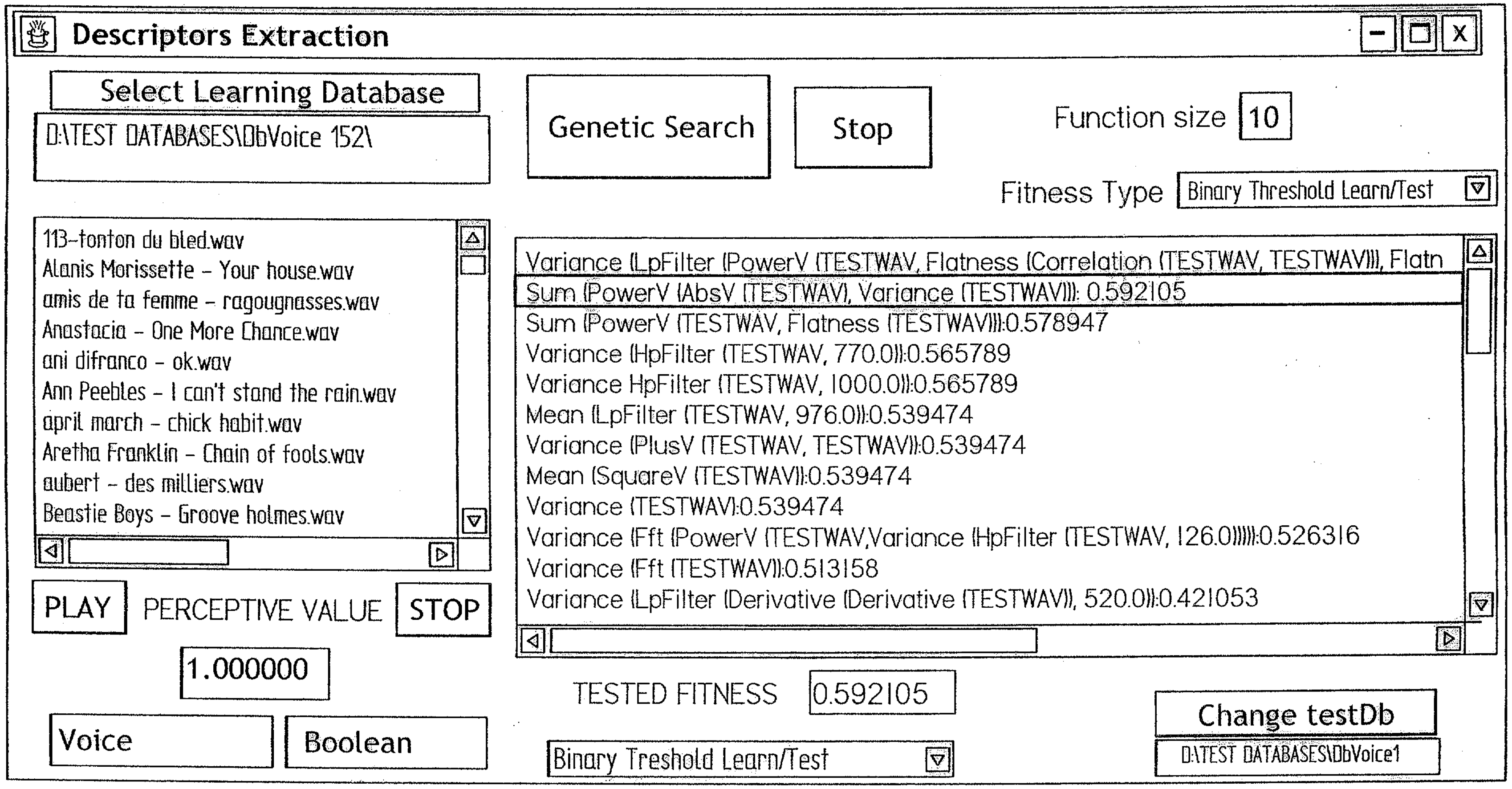

S2
Prepare learning DB 10 contng. S1⇔ Dgt1 ...Sm⇔Dgtm,
S4
Prep. population P1{CF1-CFr} of compound functions CFi from EFs
S6
Modify current pop. using CFs by manipulating their EFs as symbolic objects
S8
Op. functions of pop. on S1, ...,Sm⇒ calcltd. value for each CF of pop. w.r.t. S1, ...Sm
S10
For each CF of pop. det. matching deg. between its calculated value Dgti for the Si from which it was calculated
S12
Select CFs of pop. producing the best matches⇒ new generation P2 of CFs.
Q1
Ending criterion satisfied ?
Yes
No
S14
new pop.⇒ current pop.
S16
Output at least one function of current new pop. as a general function (4)

FIG.12



D:\TEST DATABASES\DbVoice 152\

Function size 10

Fitness Type Binary Threshold Learn/Test

113-tonton du bled.wav
Alanis Morissette - Your house.wav
amis de ta femme - ragougnasses.wav
Anastacia - One More Chance.wav
ani difranco - ok.wav
Ann Peebles - I can't stand the rain.wav
april march - chick habit.wav
Aretha Franklin - Chain of fools.wav
aubert - des milliers.wav
Beastie Boys - Groove holmes.wav

```
Variance (LpFilter (PowerV (TESTWAV, Flatness (Correlation (TESTWAV, TESTWAV))), Flatn
Sum (PowerV (AbsV (TESTWAV), Variance (TESTWAV)): 0.592105
Sum (PowerV (TESTWAV, Flatness (TESTWAV))):0.578947
Variance (HpFilter (TESTWAV, 770.0)):0.565789
Variance HpFilter (TESTWAV, 1000.0)):0.565789
Mean (LpFilter (TESTWAV, 976.0)):0.539474
Variance (PlusV (TESTWAV, TESTWAV)):0.539474
Mean (SquareV (TESTWAV)):0.539474
Variance (TESTWAV):0.539474
Variance (Fft (PowerV (TESTWAV,Variance (HpFilter (TESTWAV, 126.0))))):0.526316
Variance (Fft (TESTWAV)):0.513158
Variance (LpFilter (Derivative (Derivative (TESTWAV)), 520.0)):0.421053
```

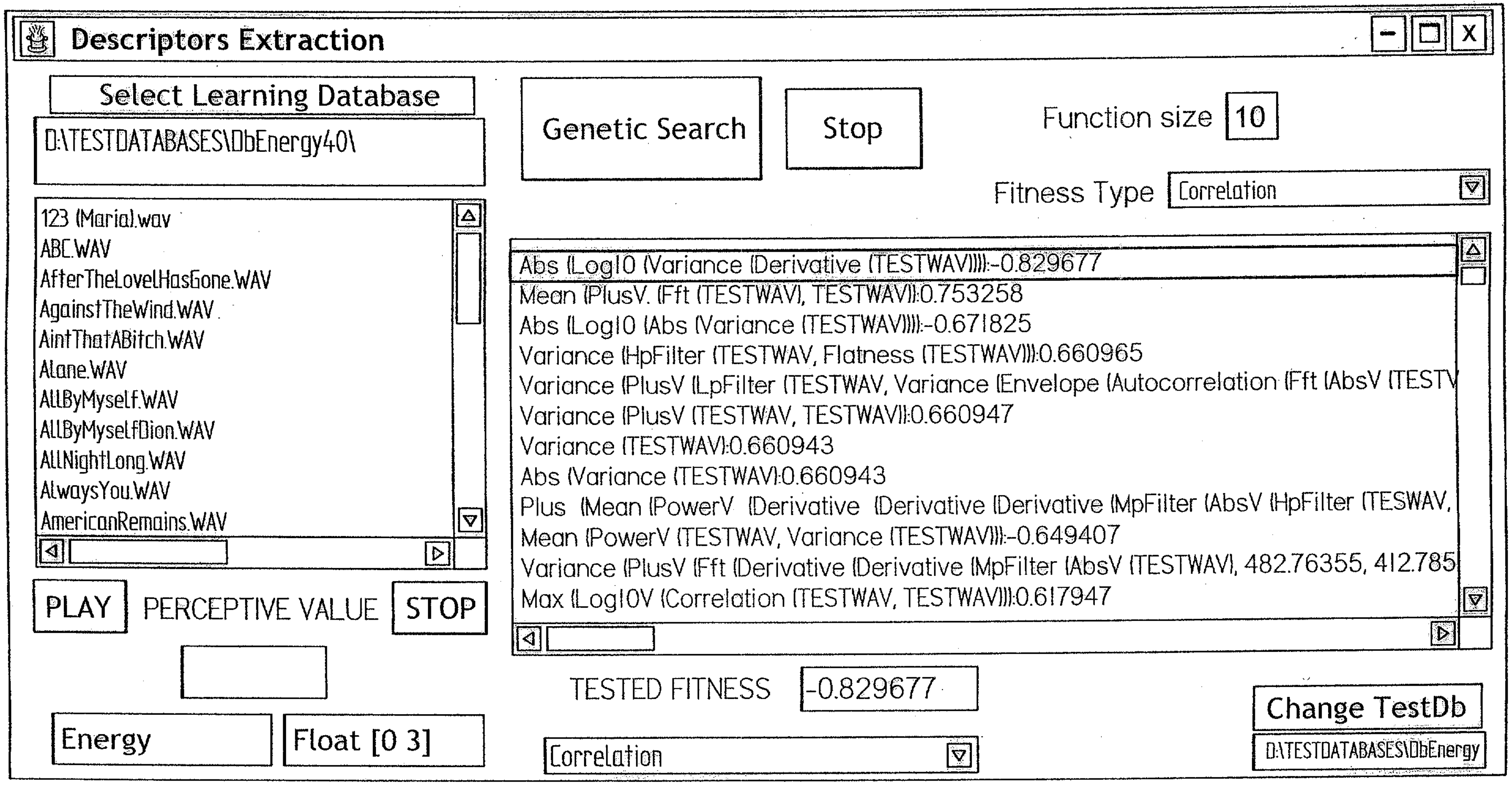
PERCEPTIVE VALUE 1.000000

TESTED FITNESS 0.592105

Binary Treshold Learn/Test

D:\TEST DATABASES\DbVoice1

FIG.13



D:\TESTDATABASES\DbEnergy40\

Function size 10

Fitness Type Correlation

```
123 (Maria).wav
ABC.WAV
AfterTheLoveIHasGone.WAV
AgainstTheWind.WAV
AintThatABitch.WAV
Alone.WAV
AllByMyself.WAV
AllByMyselfDion.WAV
AllNightLong.WAV
AlwaysYou.WAV
AmericanRemains.WAV
```

```
Abs (Log10 (Variance (Derivative (TESTWAV)))):-0.829677
Mean (PlusV. (Fft (TESTWAV), TESTWAV)):0.753258
Abs (Log10 (Abs (Variance (TESTWAV)))):-0.671825
Variance (HpFilter (TESTWAV, Flatness (TESTWAV))):0.660965
Variance (PlusV (LpFilter (TESTWAV, Variance (Envelope (Autocorrelation (Fft (AbsV (TESTV
Variance (PlusV (TESTWAV, TESTWAV)):0.660947
Variance (TESTWAV):0.660943
Abs (Variance (TESTWAV):0.660943
Plus  (Mean (PowerV  (Derivative  (Derivative (Derivative (MpFilter (AbsV (HpFilter (TESWAV,
Mean (PowerV (TESTWAV, Variance (TESTWAV))):-0.649407
Variance (PlusV (Fft (Derivative (Derivative (MpFilter (AbsV (TESTWAV), 482.76355, 412.785
Max (Log10V (Correlation (TESTWAV, TESTWAV))):0.617947
```

Energy

Float [0 3]

TESTED FITNESS -0.829677

Correlation

D:\TESTDATABASES\DbEnergy

FIG.14

European Patent Office

EUROPEAN SEARCH REPORT

Application Number
EP 02 29 3122

DOCUMENTS CONSIDERED TO BE RELEVANT

Category	Citation of document with indication, where appropriate, of relevant passages	Relevant to claim	CLASSIFICATION OF THE APPLICATION (Int.Cl.7)
A	TOKUMARU M ET AL: "MEMBERSHIP FUNCTIONS IN AUTOMATIC HARMONIZATION SYSTEM" PROCEEDINGS OF THE 1998 28TH IEEE INTERNATIONAL SYMPOSIUM ON MULTIPLE-VALUED LOGIC. ISMVL '98. FUKUOKA, MAY 27 - 29, 1998, THE INTERNATIONAL SYMPOSIUM ON MULTIPLE-VALUED LOGIC, LOS ALAMITOS, CA: IEEE COMPUTER SOC, US, 27 May 1998 (1998-05-27), pages 350-355, XP000793476 ISBN: 0-8186-8372-4 * page 352, left-hand column, line 15 - page 355, left-hand column * ---	1-29	G10H1/00
A	LAMBROU T ET AL: "CLASSIFICATION OF AUDIO SIGNALS USING STATISTICAL FEATURES ON TIME AND WAVELET TRANSFORM DOMAINS" PROCEEDINGS OF THE 1998 IEEE INTERNATIONAL CONFERENCE ON ACOUSTICS, SPEECH AND SIGNAL PROCESSING. ICASSP '98. SEATTLE, WA, MAY 12 - 15, 1998, IEEE INTERNATIONAL CONFERENCE ON ACOUSTICS, SPEECH AND SIGNAL PROCESSING, NEW YORK, NY: IEEE, US, vol. 6 CONF. 23, 12 May 1998 (1998-05-12), pages 3621-3624, XP000951242 ISBN: 0-7803-4429-4 * page 3621, left-hand column, line 21 - page 3623, left-hand column, line 5 * ---	1-29	TECHNICAL FIELDS SEARCHED (Int.Cl.7) G10H
A	US 6 028 262 A (MINAMITAKA JUNICHI) 22 February 2000 (2000-02-22) * column 1, line 41 - column 2, line 53 * * column 4, line 57 - column 6, line 14 * * figures 4-6,8 * --- -/--	1-29	

The present search report has been drawn up for all claims

Place of search	Date of completion of the search	Examiner
THE HAGUE	22 May 2003	Pulluard, R

CATEGORY OF CITED DOCUMENTS

X : particularly relevant if taken alone
Y : particularly relevant if combined with another document of the same category
A : technological background
O : non-written disclosure
P : intermediate document

T : theory or principle underlying the invention
E : earlier patent document, but published on, or after the filing date
D : document cited in the application
L : document cited for other reasons

& : member of the same patent family, corresponding document

European Patent Office

EUROPEAN SEARCH REPORT

Application Number
EP 02 29 3122

DOCUMENTS CONSIDERED TO BE RELEVANT

Category	Citation of document with indication, where appropriate, of relevant passages	Relevant to claim	CLASSIFICATION OF THE APPLICATION (Int.Cl.7)
A	US 5 210 820 A (KENYON STEPHEN C) 11 May 1993 (1993-05-11) * column 16, line 13 - column 17, line 30 * -----	1,18-29	
			TECHNICAL FIELDS SEARCHED (Int.Cl.7)

The present search report has been drawn up for all claims

Place of search	Date of completion of the search	Examiner
THE HAGUE	22 May 2003	Pulluard, R

CATEGORY OF CITED DOCUMENTS

X : particularly relevant if taken alone
Y : particularly relevant if combined with another document of the same category
A : technological background
O : non-written disclosure
P : intermediate document

T : theory or principle underlying the invention
E : earlier patent document, but published on, or after the filing date
D : document cited in the application
L : document cited for other reasons
& : member of the same patent family, corresponding document

ANNEX TO THE EUROPEAN SEARCH REPORT ON EUROPEAN PATENT APPLICATION NO. EP 02 29 3122

This annex lists the patent family members relating to the patent documents cited in the above-mentioned European search report. The members are as contained in the European Patent Office EDP file on
The European Patent Office is in no way liable for these particulars which are merely given for the purpose of information.

22-05-2003

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 6028262 A	22-02-2000	JP 11231871 A	27-08-1999
US 5210820 A	11-05-1993	AT 142815 T	15-09-1996
		CA 2041754 A1	03-11-1991
		DE 69122017 D1	17-10-1996
		DE 69122017 T2	10-04-1997
		EP 0480010 A1	15-04-1992
		ES 2091328 T3	01-11-1996
		HK 133697 A	24-10-1997
		JP 5501166 T	04-03-1993
		JP 3130926 B2	31-01-2001
		WO 9117540 A1	14-11-1991

For more details about this annex : see Official Journal of the European Patent Office, No. 12/82