



(12)

EUROPEAN PATENT APPLICATION

(43)

Date of publication:  
28.08.2024 Bulletin 2024/35

(51)

International Patent Classification (IPC):  
H04S 7/00 (2006.01)

(21)

Application number: 23305234.9

(52)

Cooperative Patent Classification (CPC):  
H04S 7/30; H04S 7/40; H04S 2400/11

(22)

Date of filing: 22.02.2023

<div>(84)</div> <div>Designated Contracting States: AL AT BE BG CH CY CZ DE DK EE ES FI FR GB GR HR HU IE IS IT LI LT LU LV MC ME MK MT NL NO PL PT RO RS SE SI SK SM TR Designated Extension States: BA Designated Validation States: KH MA MD TN</div> <div>(71)</div> <div>Applicant: L-Acoustics 91462 Marcoussis Cedex (FR)</div>	<div>(72)</div> <div>Inventors: • BENILOV, Arthur LONDON, UB6 8NS (GB) • LE NOST, Guillaume LONDON, SE4 2PB (GB)</div> <div>(74)</div> <div>Representative: Novagraaf Technologies Bâtiment O2 2, rue Sarah Bernhardt CS90017 92665 Asnières-sur-Seine Cedex (FR)</div>
--	---

(54)

METHOD AND DEVICE FOR PROCESSING AUDIO OBJECTS

(57)

A method for generating at least one output audio object, comprising: obtaining (710) a first group of input audio objects associated with respective input audio signals and input object metadata; allowing (720) a user to define a first function chain including a group analysis function and a metadata generation function, wherein the group analysis function is configured to perform a statistical and / or comparative analysis of input audio

objects in the first group to generate first analysis meta-  
data, wherein the metadata generation function is con-  
figured to generate first output object metadata for the at  
least one output audio object based on at least one meta-  
data generation parameter computed based on at least  
one part of the first analysis metadata; applying (730) the  
first function chain to the first group of input audio objects.

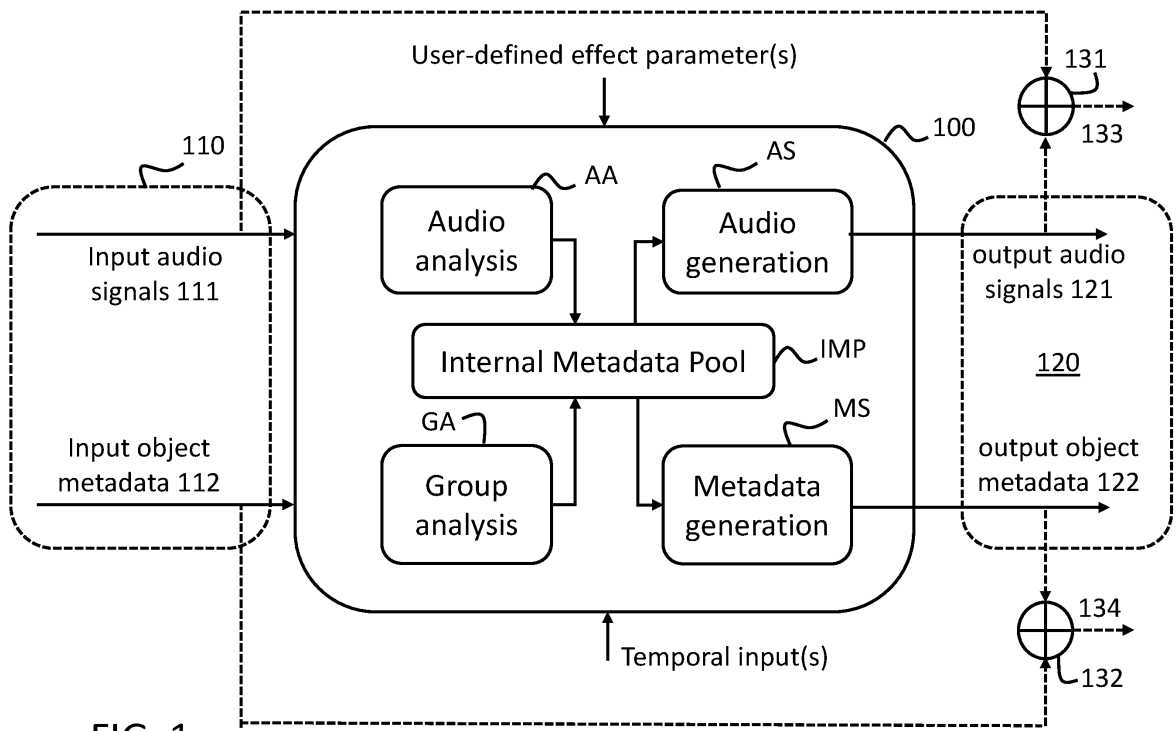


FIG. 1

EP 4 422 214 A1

**Description**TECHNICAL FIELD

**[0001]** Various example embodiments relate generally to audio object processing. More precisely a method for processing audio objects based on an extended audio effect engine and a corresponding device are disclosed.

BACKGROUND

**[0002]** Object-based audio processing devices include two large families of systems : the Digital Audio Workstations (or DAWs), and the live processing devices (such as live and broadcast mixing consoles or live spatial audio processors).

**[0003]** In such object-based audio devices, an audio object is defined as:

- A mono or stereo audio asset defining audio signal(s) (based on either an audio file or a live audio signal source);
- Metadata describing the audio object: name, spatial position (either in cartesian or spherical coordinates), spatial size, and any other useful metadata (audio gain, etc.)

**[0004]** In a DAW, N audio objects are usually represented in a user interface by N audio tracks. The track for a given audio object is associated with audio signal data and metadata. These metadata may be adjusted either via a panning plugin or via a native panner. The DAW provides a timeline on which the evolution of each metadata (such as the object 3D coordinates) can be recorded, edited and played back.

**[0005]** In a live processing device, the notion of audio file is replaced by the notion of live audio source (from a stage microphone, for example), and the notion of timeline is replaced by the notion of snapshots and timecode. Snapshots store given values of the objects metadata and can be triggered when a specific timecode is reached.

**[0006]** In these two types of audio processing devices, audio objects evolve independently from each other. The mixing engineer can record trajectories or apply effects to an object via automation or even via effects such as low frequency oscillators applied to a particular metadata (example : change X in a back-and-forth movement for object (k)).

**[0007]** To save editing time, some audio processing devices usually provide a functionality for groups of objects that the user can define and select to edit/record the position of several objects at the same time. However, apart from a mirror mode for stereo panning, there is no way to define a complex relational behaviour for the audio objects belonging to a group of objects.

**[0008]** An objective is to provide a higher level of creation and control for differentiated motions or audio sig-

nals generation for a plurality of audio objects.

SUMMARY

**[0009]** The scope of protection is set out by the independent claims. The embodiments, examples and features, if any, described in this specification that do not fall under the scope of the protection are to be interpreted as examples useful for understanding the various embodiments or examples that fall under the scope of protection.

**[0010]** According to a first aspect, a method for generating at least one output audio object is disclosed. The method comprising: obtaining a first group of input audio objects associated with respective input audio signals and input object metadata; allowing a user to define a first function chain including a group analysis function and a metadata generation function, wherein the group analysis function is configured to perform a statistical and / or comparative analysis of input audio objects in the first group to generate first analysis metadata, wherein the metadata generation function is configured to generate first output object metadata for the at least one output audio object based on at least one metadata generation parameter computed based on at least one part of the first analysis metadata; applying the first function chain to the first group of input audio objects, wherein applying the first function chain includes: generating the first analysis metadata by applying the group analysis function to the input audio objects in the first group; generating the first output object metadata for the at least one output audio object using the metadata generation function.

**[0011]** In some embodiments, the method comprises: storing a metadata pool associated with the first function chain, the metadata pool including the first analysis metadata and the input object metadata; allowing the user to define at least one metadata generation parameter of the metadata generation function based on any of the metadata stored in the metadata pool.

**[0012]** In some embodiments, the first function chain includes an audio generation function configured to generate at least one output audio signal for the at least one output audio object; the method comprising: allowing a user to define the audio generation function having at least one audio generation parameter computed based on at least one part of the first analysis metadata; wherein applying the first function chain includes generating the at least one output audio signal using the audio generation function.

**[0013]** In some embodiments, the metadata generation function is configured to generate the output object metadata based on at least part of the input object metadata.

**[0014]** In some embodiments, the audio generation function is configured to generate the at least one output audio signal based on at least one of the input audio signals.

**[0015]** In some embodiments, the first function chain

includes an audio analysis function configured to perform an analysis of at least one audio signals to generate one or more audio signal metrics, the method comprising: allowing a user to define the audio analysis function; wherein applying the first function chain includes generating one or more first audio signal metrics using the audio analysis function for performing an analysis of at least one of the input audio signals, the metadata pool including the one or more first audio signal metrics.

**[0016]** In some embodiments, the first analysis metadata includes a statistical parameter; wherein applying the group analysis function includes performing a statistical analysis of the input audio objects in the first group to generate the statistical parameter.

**[0017]** In some embodiments, the first analysis metadata includes a comparative parameter; wherein applying the group analysis function includes performing a comparative analysis of metadata of the input audio objects in the first group to generate the comparative parameter.

**[0018]** In some embodiments, the metadata generation function has at least one parameter computed based on time.

**[0019]** In some embodiments, the metadata generation function has at least one parameter computed based on a user-defined temporal event.

**[0020]** In some embodiments, the audio generation function has at least one parameter computed based on a user-defined temporal event.

**[0021]** In some embodiments, the method comprises: obtaining a second group of input audio objects associated with respective second input audio signals and second input object metadata, wherein the second group includes the at least one output audio object; allowing a user to define a second function chain including a second group analysis function and a second metadata generation function, wherein the second group analysis function is configured to perform a statistical and / or comparative analysis of input audio objects in the second group to generate second analysis metadata, wherein the metadata generation function is configured to generate second output object metadata for at least one second output audio object and has at least one second metadata generation parameter computed based on at least one part of the second analysis metadata; applying the second function chain to the second group of input audio objects, wherein applying the second function chain includes: generating the second analysis metadata by applying the second group analysis function to the input audio objects in the second group; generating the second output object metadata for at least one second output audio object using the second metadata generation function.

**[0022]** In some embodiments, the method comprises: storing a second metadata pool associated with the second function chain, the metadata pool including the second analysis metadata and the second input object metadata; allowing the user to define at least one metadata generation parameter of the metadata generation func-

tion based on any of the metadata stored in the second metadata pool.

**[0023]** In some embodiments, the method comprises: allowing a user to define a second audio generation function that is configured to generate at least one audio signal and has at least one second audio generation parameter computed based on at least one part of the second analysis metadata; generating at least one second output audio signal for the at least one second output audio object using the second audio generation function.

**[0024]** According to another aspect, a device comprises means for performing a method comprising: obtaining a first group of input audio objects associated with respective input audio signals and input object metadata; allowing a user to define a first function chain including a group analysis function and a metadata generation function, wherein the group analysis function is configured to perform a statistical and / or comparative analysis of input audio objects in the first group to generate first analysis metadata, wherein the metadata generation function is configured to generate first output object metadata for the at least one output audio object based on at least one metadata generation parameter computed based on at least one part of the first analysis metadata; applying the first function chain to the first group of input audio objects, wherein applying the first function chain includes: generating the first analysis metadata by applying the group analysis function to the input audio objects in the first group; generating the first output object metadata for the at least one output audio object using the metadata generation function.

**[0025]** The device may comprise means for performing one or more or all steps of the method according to the first aspect. The means may include circuitry configured to perform one or more or all steps of a method according to the first aspect. The means may include at least one processor and at least one memory storing instructions that, when executed by the at least one processor, cause the device to perform one or more or all steps of a method according to the first aspect.

**[0026]** According to another aspect, a device comprises at least one processor and at least one memory storing instructions that, when executed by the at least one processor, cause the device to perform: obtaining a first group of input audio objects associated with respective input audio signals and input object metadata; allowing a user to define a first function chain including a group analysis function and a metadata generation function, wherein the group analysis function is configured to perform a statistical and / or comparative analysis of input audio objects in the first group to generate first analysis metadata, wherein the metadata generation function is configured to generate first output object metadata for the at least one output audio object based on at least one metadata generation parameter computed based on at least one part of the first analysis metadata; applying the first function chain to the first group of input audio objects, wherein applying the first function chain includes: generating the

first analysis metadata by applying the group analysis function to the input audio objects in the first group; generating the first output object metadata for the at least one output audio object using the metadata generation function.

**[0027]** The instructions, when executed by the at least one processor, may cause the device to perform one or more or all steps of a method according to the first aspect.

**[0028]** According to another aspect, a computer program comprises instructions that, when executed by a device, cause the device to perform: obtaining a first group of input audio objects associated with respective input audio signals and input object metadata; allowing a user to define a first function chain including a group analysis function and a metadata generation function, wherein the group analysis function is configured to perform a statistical and / or comparative analysis of input audio objects in the first group to generate first analysis metadata, wherein the metadata generation function is configured to generate first output object metadata for the at least one output audio object based on at least one metadata generation parameter computed based on at least one part of the first analysis metadata; applying the first function chain to the first group of input audio objects, wherein applying the first function chain includes: generating the first analysis metadata by applying the group analysis function to the input audio objects in the first group; generating the first output object metadata for the at least one output audio object using the metadata generation function.

**[0029]** The instructions may cause the device to perform one or more or all steps of a method according to the first aspect.

**[0030]** According to another aspect, a non-transitory computer readable medium comprises program instructions stored thereon for causing a device to perform at least the following: obtaining a first group of input audio objects associated with respective input audio signals and input object metadata; allowing a user to define a first function chain including a group analysis function and a metadata generation function, wherein the group analysis function is configured to perform a statistical and / or comparative analysis of input audio objects in the first group to generate first analysis metadata, wherein the metadata generation function is configured to generate first output object metadata for the at least one output audio object based on at least one metadata generation parameter computed based on at least one part of the first analysis metadata; applying the first function chain to the first group of input audio objects, wherein applying the first function chain includes: generating the first analysis metadata by applying the group analysis function to the input audio objects in the first group; generating the first output object metadata for the at least one output audio object using the metadata generation function.

**[0031]** The program instructions may cause the device to perform one or more or all steps of a method according to the first aspect.

## BRIEF DESCRIPTION OF THE DRAWINGS

**[0032]** Example embodiments will become more fully understood from the detailed description given herein below and the accompanying drawings, which are given by way of illustration only and thus are not limiting of this disclosure.

FIG. 1 is a schematic diagram illustrating an audio object processing unit according to an example.

FIG. 2A is a schematic diagram illustrating an example of a chain of effects.

FIG. 2B is a schematic diagram illustrating an example of a chain of effects.

FIG. 3 is a schematic diagram illustrating an audio object processing unit for implementing a spatial randomizer.

FIG. 4 is a schematic diagram illustrating an audio object processing unit for implementing an attraction effect.

FIG. 5 is a schematic diagram illustrating an audio object processing unit for implementing a spatial audio synthesizer.

FIG. 6 shows an example of user interface allowing to define a function chain for defining for defining an audio object processing unit according to an example.

FIG. 7 shows a flowchart of a method for generating at least one output audio object from a group of input audio objects according to an example.

FIG. 8 is a block diagram illustrating an exemplary hardware structure of a processing device according to an example.

**[0033]** It should be noted that these drawings are intended to illustrate various aspects of devices, methods and structures used in example embodiments described herein. The use of similar or identical reference numbers in the various drawings is intended to indicate the presence of a similar or identical element or feature.

## DETAILED DESCRIPTION

**[0034]** Detailed example embodiments are disclosed herein. However, specific structural and/or functional details disclosed herein are merely representative for purposes of describing example embodiments and providing a clear understanding of the underlying principles. However these example embodiments may be practiced without these specific details. These example embodiments may be embodied in many alternate forms, with various modifications, and should not be construed as limited to only the embodiments set forth herein. In addition, the figures and descriptions may have been simplified to illustrate elements and / or aspects that are relevant for a clear understanding of the present invention, while eliminating, for purposes of clarity, many other elements that may be well known in the art or not relevant for the un-

derstanding of the invention.

**[0035]** An extended audio effect engine is disclosed that provides a high level of creation and control for differentiated motions or signals generations for a plurality of audio objects. The extended audio effect engine provides audio object processing units based on a chain of functions to a user to help him/her define an overall spatial behaviour and/or the audio signals of the audio objects in an audio scene.

**[0036]** The extended audio effect engine allows to define rules and / or relations between objects such as « if this object A is at this position, this other object B should move a bit further away ». Effects such as « to control the behaviour of audio objects as a flock of birds » can be implemented.

**[0037]** Creative use-cases can be generated, where temporal behaviours and / or metadata modulation such as spatial motions or additional audio signals synthesis in a deterministic or non-deterministic manner, for a group of audio objects, from temporal events and/or user-defined parameters can be defined.

**[0038]** A device and corresponding method for generating at least one output audio object from a group of input audio objects using an audio object processing unit based on a function chain is disclosed. The group of input audio objects are associated with respective input audio signals and input object metadata.

**[0039]** The audio object processing unit uses the group of input audio objects as input and generates at least one output audio object as output. The audio object processing unit allows to define a function chain to generate one or more output audio object(s). The function chain may correspond either to an effect configured to be applied and modify one or more of the input audio objects to generate one or more output audio object(s) or a synthesis function configured to generate one or more new output audio object(s) based on metrics derived from the input audio objects.

**[0040]** In the context of this document, the terms "effect" or "processing unit" may also be used to designate the audio object processing unit.

**[0041]** The audio object processing unit allows a user to define the function chain based on several functions and relationships between these functions. These functions link the spatial behaviour and/or audio signals of the group of input audio objects to the output audio object(s). These functions include a group analysis function and metadata generation function. These functions may also include an audio analysis function and an audio generation function.

**[0042]** The group analysis function is configured to perform a statistical and / or comparative analysis of input audio objects in the group of input audio objects to generate "analysis metadata".

**[0043]** The terms "analysis metadata" are used herein to designate output data generated by an analysis function (either a group analysis function or an audio analysis function). The "analysis metadata" may include analysis

metrics (e.g. statistical parameters), i.e. one or more values of one or more parameters that are the object of the analysis performed by the concerned analysis function.

**[0044]** The metadata generation function is configured to generate output object metadata 122 for one or more output audio objects 120 and has at least one metadata generation parameter (e.g. input parameter and/or function parameter) computed based on at least one part of the analysis metadata. The metadata generation function may apply an effect to existing metadata of existing audio objects (e.g. the metadata of one or more input audio objects in the group 110 of input audio objects) to generate output object metadata 122 for one or more output audio objects 120 or synthesize the output object metadata 122 for new output audio objects 120 without starting from existing metadata of existing audio objects.

**[0045]** The audio generation function is configured to generate output audio signal(s) 121 for one or more output audio object(s) 120 and has at least one audio generation parameter (e.g. input parameter and/or function parameter) computed based on at least one part of the analysis metadata. The audio generation function may use and apply an effect to existing input audio signal(s) (e.g. the audio signal(s) of one or more input audio objects in the group of input audio objects) or synthesize the output audio signal(s) 121 without starting from existing input audio signal(s) of existing audio objects.

**[0046]** The function chain is applied to the group of input audio objects to generate one or more output audio object(s) 120 including associated metadata 122 and output audio signal(s) 121.

**[0047]** Applying the function chain may include: generating analysis metadata using the group analysis function for performing statistical and / or comparative analysis of input audio objects in the group of input audio objects; and generating output object metadata for output audio object(s) using the metadata generation function, wherein at least one metadata generation parameter is computed based on at least one part of the analysis metadata generated by the group analysis function for the one or more input audio objects in the group.

**[0048]** Applying the function chain may further include: generating at least one output audio signal 121 for output audio object(s) 120 using the audio generation function, wherein at least one audio generation parameter is computed based on at least one part of the analysis metadata generated by the group analysis function for the one or more input audio objects in the group.

**[0049]** FIG. 1 is a schematic diagram showing functional blocks of a processing unit 100 configured to define and run a function chain according to an example.

**[0050]** The processing unit 100 includes:

- A group analysis block (GA) performing a group analysis function;
- A metadata generation block (MS) performing a metadata generation function.

**[0051]** The processing unit 100 may further include:

- An audio analysis block (AA) performing an audio analysis function;
- An audio generation block (AS) performing an audio generation function.

**[0052]** The processing unit 100 receives as input a group 110 of input audio objects associated with respective input audio signals 111 and input object metadata 112. The group may be a user-defined plurality of audio objects. One processing unit may relate to one group of input audio objects, meaning that the processing unit may be defined specifically for this group of input audio objects.

**[0053]** In the context of the present disclosure, the words "group" (of elements, e.g. audio objects), "plurality" (of elements, e.g. audio objects), "pool" (of elements, here metadata pool) have the same meaning and are used to designate a set of elements.

**[0054]** The processing unit 100 may be configured to receive the metadata 112 of the input audio objects (referred to herein as the input object metadata) and associated input audio signals 111. The metadata of an audio object may include two types of metadata: spatial metadata (e.g. position, spatial coordinates, object size, etc) related to the spatial behaviour of the audio object and audio metadata (e.g. audio RMS (root mean square) gain, amplitude peak level, audio spectral magnitude, etc) related to the audio signal associated with the audio object. The spatial coordinates may be defined in a cartesian (XYZ) or spherical (AED) model.

**[0055]** The processing unit 100 may further be configured to receive user-defined effect parameters, corresponding to effect parameters defined by a user, e.g. independently of the input audio objects.

**[0056]** The processing unit 100 may further be configured to receive temporal inputs, in the form of timecode, time length, clock signal, or temporal sequence of events (such as MIDI notes or MIDI control messages).

**[0057]** The processing unit 100 is configured to generate, store and use an internal metadata pool (IMP). The IMP is a pool of metadata associated with the function chain defined by means of the processing unit. The metadata generation function MS and/or the audio generation function AS are chained with the group analysis function GA and/or the audio analysis function AA through the internal metadata pool.

**[0058]** The IMP may include analysis metadata generated by the group analysis function GA and/or the audio analysis function AA. The IMP may include other metadata that may be needed by the metadata generation function MG and/or the audio generation function AS. The IMP may include the following metadata:

- spatial and / or audio metadata of the input audio objects within the group (such as XYZ coordinates or size);

- any other metadata related to the input audio objects within the group (e.g. name, type, etc.)
- one or more analysis metadata (e.g. statistical parameter(s) or comparative parameter(s)) generated by the group analysis function (GA);
- one or more analysis metadata (e.g. audio signal metric(s)) generated by the audio analysis function (AA).

**[0059]** A metadata value stored in the IMP is associated with a given parameter that may be static or dynamic. For a static parameter having a fixed value over time, the IMP include a value of the static parameter. For a dynamic parameter, the IMP includes values of the dynamic parameter varying as a function of time.

**[0060]** The Internal Metadata Pool, IMP is fed by the audio analysis and group analysis blocks. The IMP may be fed in real-time and/or dynamically based on the audio signals and / or dynamically varying metadata of the input audio objects. This IMP allows for value(s) of a parameter computed for an input audio object in the group to influence either another input audio object in the group or a newly synthesized output audio object.

**[0061]** The group analysis function (GA) generates or updates the metadata in the IMP. This GA function may be configured to implement a mathematical analysis (for example, a statistical analysis) of input objects of the group to generate analysis metadata, including for example one or more analysis metrics. The group analysis function may perform the analysis based on one or more metadata of the input objects in the group.

**[0062]** Any type of metric that could be useful for the synthesis blocks may be generated by the group analysis function. A metric may be a static metric, having a fixed value over time or a dynamic metric, varying as a function of time. A metric may be:

- a statistical parameter (e.g. a statistical parameter computed for input audio objects in the group of input audio objects, such parameter may be referred to as an objects group metric) such as number of input audio objects in the group, another statistical parameter (mean, standard deviation, etc) computed for a given metadata (e.g. position, gain, etc) over the group; or
- a comparative parameter (e.g. a parameter comparing two or more input audio objects on the basis of one or more parameters in the input object metadata, such parameter may be referred to as an inter-object metric) such as for example but not limited to: spatial distance between input audio objects, difference(s) computed within the group for a specific Boolean metadata linked to a user effect parameter (such as, « is the object part of a stereo pair »), deltaX (a difference between cartesian coordinates) or deltaAzimuth (a difference between spherical coordinates), or type comparison.

**[0063]** The audio analysis function (AA) generates or updates the metadata in the IMP. This AA function may be configured to implement a signal processing analysis on one or more audio signals associated with one or more input audio objects in the group to generate analysis metadata, including for example one or more audio signal metrics. An audio analysis function may include any type of signal analysis function applicable to an audio signal including but not limited to: RMS Level, peak level, band-limited Level, audio features extraction using FFT.

**[0064]** The processing unit 100 relies on a set of processing parameters including the IMP. The set of processing parameters may include externally generated parameters: the temporal inputs of the processing unit and/or the user-defined effect parameters. This set of processing parameters is used to define one or more parameters of the audio generation function and/or the metadata generation function.

**[0065]** The processing unit 100 provides a user interface configured to allow a user to define each of the group analysis function, audio analysis function, metadata generation function and audio generation function that is part of a given processing unit. Each of these functions may be defined by a user by:

- selecting a predefined function within a catalogue of predefined functions; and / or
- providing user input(s) that define the type of function; and /or
- providing a mathematical expression of the function.

**[0066]** Several parameters are associated with each function. Each function may have one or more input parameters to which the function is applied. Each function may have one or more output parameters generated as output. The behavior of the function may be defined by one or more function parameter. For example, an audio analysis function that computes as output parameter a weighted sum of audio gains of audio signals uses as input parameters the audio gains and as function parameters the weights of the weighted sum.

**[0067]** Each input and/or function parameter (referred to herein as an audio analysis parameter, a group analysis parameter, an audio generation parameter, a metadata generation parameter) associated with a function (audio analysis function, group analysis function, an audio generation function or respectively a metadata generation function) may be defined by a user by:

- selecting one or more parameters in the IMP and / or selecting one or more user-defined parameters and / or selecting a user-defined temporal event and / or selecting a time input (timestamp, time step);
- selecting the parameter in a list of predefined parameters, e.g. for a predefined function;
- expressing a mathematical relationship between the selected parameter(s) and the parameter to be defined (e.g. by default if only one parameter is select-

ed, the defined parameter is equal to the selected parameter);

- optionally naming the parameter;
- optionally providing a range of values for the parameter;
- etc.

**[0068]** The processing unit 100 provides a user interface configured to allow a user to define one or more parameters (hereafter, function parameter) of the considered function. The user interface may allow the user to define at least one function parameter based on any of the metadata stored in the IMP or in the set of processing parameters. A function parameter may be defined by selecting a parameter in the set of processing parameters (e.g. in the IMP). A function parameter may further be defined by defining the mathematical relationships between the parameter selected in the set of processing parameters and the function parameter. If the parameter selected in the set of processing parameters is equal to the function parameter, no mathematical relationship needs to be entered.

**[0069]** To facilitate the identification of the parameters in the set of processing parameters, a name or identifier may be given (automatically or by the user) to the analysis metadata (e.g. analysis metrics) generated respectively by the group analysis function or audio analysis function. A parameter of metadata generation function or audio generation function may thus be defined by selecting in the set of processing parameters one or more of the named analysis metadata generated by the group analysis function or audio analysis function and / or by selecting other parameters in the set of processing parameters.

**[0070]** The metadata generation function MS is configured to generate output object metadata 122 for the output audio objects 120. The metadata generation function has one or more metadata generation parameters defined based one or more effect parameter from the set of processing parameters. The metadata generation function may for example generate spatial metadata defining deterministic or non-deterministic individualized motions for the output audio objects 120, based on time and/or temporal events and/or the IMP.

**[0071]** As illustrated by FIG. 1, a "modulation" may be implemented for input audio signals: an output audio signal 121 may be added to an input audio signal 111 to generate a modified ("modulated") audio signal 133. In this embodiment, the modulation is an additive modulation. Multiplicative modulation may also be considered. An output audio signal 121 may define relative variations (e.g. relative amplitude variations) for a corresponding input audio signal. The output audio signal 121 may be added (using an adder 131) to each input audio signal 111 to be modulated. Several input audio signals may be modulated in this manner, using a same or several output audio signals 121. In one or more alternative embodiments, the modulation of audio signals is implement-

ed by the audio generation function AS.

**[0072]** Likewise, a "modulation" may be implemented for input metadata: an output metadata signal 122 (corresponding to values of a dynamic parameter in the output object metadata 122) may be added to a metadata input signal 112 (corresponding to one or more values of a static or dynamic parameter in the input object metadata 112) to generate a modified ("modulated") metadata signal 134.. In this embodiment, the modulation is an additive modulation. Multiplicative modulation may also be considered. An output metadata signal 122 may define relative variations (e.g. relative metadata) for the input metadata signal 112. The metadata output signal 122 may be added (using an adder 132) to each metadata input signal 112 to be modulated. Several metadata input signals may be modulated in this manner, using a same or several metadata output signals 122. In one or more alternative embodiments, the modulation on metadata is implemented by the metadata generation function MS.

**[0073]** The metadata generation function MS may for example be configured to generate spatial modulations (i.e. relative positioning information) for the input audio objects within the group. For each input audio object  $k$ , and each metadata  $m_k$ , a spatial modulation is defined for the metadata generation function based on a mathematical function  $f$  of the following form:

$$m_k = f(k, t, e, IMP)$$

where  $k$  represents the input object index within the group,  $t$  represents time,  $e$  represents an event, and IMP is the Internal Metadata Pool. The function  $f$  may be deterministic or non-deterministic.

**[0074]** The audio generation function (AS) generates one or more output audio signals 121 associated respectively to one or more output audio objects 120. The audio generation function has one or more audio generation parameters defined based one or more effect parameter from the set of processing parameters. The generated output audio signals 121 can be generated based on zero audio signal (i.e. the output audio signals are new synthesized audio signals), by applying effect(s) to one or more input audio signals.

**[0075]** Audio object processing units (e.g. effects) can be chained to generate complex motions or complex audio signals for one or more groups of objects, as illustrated by the chain of effects represented by FIG. 2.

**[0076]** In the example of FIG. 2A, a first processing unit 210 corresponding a first effect 210 is chained with a second processing unit 230 corresponding a second effect 230, each processing unit 210, 230 may implement the same function chain or a respective function chain. The description made by reference to FIG. 1 of the processing unit 100 is applicable to the first processing unit 210 and the second processing unit 230.

**[0077]** The first effect 210 receives a first group 200 of input audio objects associated with respective input au-

dio signals 201 and input object metadata 202 and generates one or more first output audio objects 220 associated with respective input audio signals 221 and input object metadata 222.

**[0078]** The second effect 230 receives a second group of input audio objects 220 associated with respective input audio signals 221 and input object metadata 222 and generate one or more second output audio objects 240 associated with respective input audio signals 241 and input object metadata 242.

**[0079]** The second group of input audio objects 220 of the second effect 230 includes some or all of the first output audio objects 220 of the first effect 210 and may include other input objects (not processed by the first effect). FIG. 2 shows the case where the second group of input audio objects corresponds to the first output audio object(s) at the output of the first effect 210.

**[0080]** The order of effects in the effect chain matters to the overall synthesis of audio and/or movements.

**[0081]** FIG. 2B shows how audio object processing units (e.g. effects) can be chained to generate when modulation of audio signals and / or metadata is used. The elements sharing the same reference signs than in FIG. 2A are identical to those described by reference to FIG. 2A and will be described again.

**[0082]** As illustrated by FIG. 2B, an audio modulation may be implemented at the output of the first audio object processing unit 210: an output audio signal 221 may be used to modulate an input audio signal 201 to generate a modulated audio signal 224 using an adder 223. Likewise, a metadata modulation may be implemented: a metadata output signal 222 may be used to modulate a metadata input signal 202 to generate a modulated metadata signal 226 using an adder 225.

**[0083]** The second group of input audio objects 220 received by the second audio object processing unit 230 may include:

- the (non-modulated) audio signal(s) 221; and/or
- the (modulated) audio signal(s) 224; and/or
- the (non-modulated) object metadata 222; and/or
- the (modulated) object metadata 226.

**[0084]** Further, an audio modulation may be implemented at the output of the second audio object processing unit 230: an output audio signal 241 may be used to modulate a (modulated or non-modulated) input audio signal 221 or 224 to generate a modulated audio signal 244 using an adder 243. Likewise, a metadata modulation may be implemented: a metadata output signal 242 may be used to modulate a (modulated or non-modulated) metadata signal 222 or 226 to generate a modulated metadata signal 246 using an adder 245.

**[0085]** FIG. 3 is a schematic diagram illustrating an example of effect, here a spatial randomizer, defined using a processing unit 300 based on a function chain according to an example. With the spatial randomizer, the audio objects have random positions varying over time.

**[0086]** The processing unit 300 includes three blocks an audio analysis block, a group analysis block and a metadata generation block implementing respectively an audio analysis function, a group analysis function and a metadata generation function.

**[0087]** The audio analysis block receives as input the audio signals 311 of a user-defined subset of objects selected within the input group 310.

**[0088]** The audio analysis block generates output metrics including an amplitude peak level for each of the selected input objects. The output metrics are stored in the IMP.

**[0089]** The group analysis block receives as input metadata 312 (referred to as the link status) indicating if some of the objects positions have been declared as "linked" by the user.

**[0090]** The group analysis block computes the number of objects within the group and generates output metrics including the number of random positions to be generated. The output metrics are stored in the IMP.

**[0091]** The group analysis function performed by the group analysis block counts the number of random positions that should be generated, according to the number of objects and their link status.

**[0092]** The metadata generation function uses as input this number of random positions. The metadata generation function uses a temporal information (time step) and beat parameters (e.g. a phase and sharpness) to spread in time the modulation values for each output audio object within the output object group. The metadata generation function uses a previous modulation value for each of these random positions to generate a current modulation value based on a random distribution. User-defined parameters may be entered by a user to define the parameters of this random distribution (e.g. amplitude, offset, phase spread). See also FIG. 6 for example parameters that may be used.

**[0093]** The metadata generation block generates output spatial metadata 322 for output audio object 320, the output spatial metadata 322 including random modulation values for each output audio object (cartesian coordinates X,Y and or Z or equivalently the spherical coordinates). The random modulation values 322 for an output audio object 320 being added by an adder 332 to the spatial position(s) 312 of a corresponding input audio object 310 to generate the randomly varying spatial positions 342 of the output audio object 320.

**[0094]** The output audio signals 321 correspond respectively to the input audio signals 311 without change.

**[0095]** The metadata generation function performed by the metadata generation block is based on a random values generator, scaled with a range of possible values for the next random positions and potentially spread in time according to one or more user-defined beat parameters (e.g. phase and sharpness).

**[0096]** FIG. 4 is a schematic diagram illustrating an example of effect, here an attraction effect, defined using a processing unit 400 based on a function chain accord-

ing to an example. An attraction effect mimics a group behaviour of individuals trying to get closer to one individual tagged as "target".

**[0097]** The processing unit 400 includes three blocks an audio analysis block, a group analysis block and a metadata generation block implementing respectively an audio analysis function, a group analysis function and a metadata generation function.

**[0098]** The audio analysis block receives as input the audio signals 411 for all input audio objects 410.

**[0099]** The audio analysis block generates output metrics including the fast RMS level. The output metrics are stored in the IMP.

**[0100]** The audio analysis function performed by the audio analysis block is short-term root mean square RMS computation.

**[0101]** The group analysis block receives as input the spatial coordinates of the input audio object, extracted from the input object metadata 412.

**[0102]** The group analysis block computes the number of objects within the group and generates output metrics including an average distance towards other input objects i.e.  $D(k) = \text{mean}(d(1,k), d(2,k), \dots, d(n,k))$  and the XYZ position of each object k amongst the n objects of the group. The output metrics are stored in the IMP.

**[0103]** The group analysis function performed by the group analysis block is the average distance towards all other objects, for each input audio object.

**[0104]** The metadata generation block uses as input for each object its spatial position, its relative distance to the other input objects, the RMS value.

**[0105]** The metadata generation block generates output spatial metadata 422 for output audio objects 420, the output spatial metadata 422 including a modulated position for each object.

**[0106]** The metadata generation function performed by the metadata generation block generates, for each audio object attracted by the "target", an update position to move towards the target object, with a speed proportional to the RMS Value, a user-defined speed parameter, and the inverse of a D(k). Update values 422 for an output audio object 420 are added by an adder 432 to the spatial position(s) 412 of a corresponding input audio object 410 to generate the updated spatial positions 442 of the output audio object 420.

**[0107]** The output audio signals 421 correspond respectively to the input audio signals 411 without change.

**[0108]** FIG. 5 is a schematic diagram illustrating an example of effect, here a spatial audio synthesizer, defined using a processing unit 500 based on a function chain according to an example. In this example, each audio object can be thought as a "voice" (i.e. a single note) of an electronic synthesiser.

**[0109]** The processing unit 500 includes three blocks, a group analysis block, an audio generation block and a metadata generation block implementing respectively a group analysis function, an audio generation function and a metadata generation function.

**[0110]** The group analysis block receives as input the spatial positions of the input audio objects 510 (in cartesian or spherical coordinates), the spatial positions being extracted from the input object metadata 512.

**[0111]** The group analysis block computes the number of objects within the group and generates output metrics including the median position and individual positions. The output metrics are stored in the IMP.

**[0112]** The metadata generation block uses as input the number of objects, a sequence of MIDI (Musical Instrument Digital Interface) events (MIDI notes), and some user parameters such as a "spatial range" that limit the spatial positions to an area.

**[0113]** The metadata generation block generates output spatial metadata 522 for output audio objects 520, the output spatial metadata 522 including a position modulation 522 for each audio object that is added to the position extracted from the input object metadata 512 of the audio object to generate a modulated position 542 at the output of the adder 532.

**[0114]** The metadata generation function performed by the metadata generation block is for example a modulation function for object k's distance to the median position, proportional to the MIDI note velocity and the spatial range. Some other creative modulation functions can be provided as a list of presets to the user.

**[0115]** The audio generation block uses as input the number of input audio objects and the MIDI events (notes and control messages).

**[0116]** The audio generation block generates output audio signals 521 for output audio objects 520, the output audio signals 521 including a synthesised note for each MIDI note received.

**[0117]** In case more notes are received simultaneously compared to the number N of audio objects defined in the processing unit 500, a selection algorithm to keep only N relevant notes is used. Notes may be assigned to object indexes by a mapping function between Note Pitch and Object Index.

**[0118]** The audio generation function performed by the audio generation block may be the additive synthesis of periodic waveforms (sinewaves, sawtooth, etc) with ADSR (Attack / Decay / Sustain / Release) envelopes and IIR (Infinite Impulse Response) filters, for each MIDI note. Some filters parameters (such as their cut-off frequency) can be mapped to MIDI control messages.

**[0119]** FIG. 6 shows an example of user interface allowing to define a function chain including a group analysis function and a metadata generation function.

**[0120]** FIG. 6 also shows an effects chain containing a randomizer effect and an attraction effect. Such a chain of effects can very efficiently simulate the behavior of a flock of birds by applying the randomizer effect (see FIG. 3) to an object k=1 (the leading bird) and applying the Attraction effect (see FIG. 4) to the other audio objects k=2 to 40. In that case, the user needs very few actions to create a complex spatial behavior for many objects.

**[0121]** On the left part 6A of the FIG. 6, a first user

interface area allows to select one effect in the chain of effects, add a new effect in the chain of effects of deleted one of the effect from the chain of effects. In FIG. 6 the randomizer effect is selected and its parameters may be adjusted.

**[0122]** In the middle part 6B of the FIG. 6, a second user interface area allows a user to represent the spatial positions and movements of the audio objects in the input audio objects. Icons 610 are available to change the viewpoint of the spatial representation.

**[0123]** In the right part 6C of the FIG. 6, a third user interface area allows a user to select the metadata to be used from the IMP and to define the parameters of the functions in the function chain of the currently selected effect. In some embodiments, the metadata selection from the IMP is done through a simple drop-down list. In others, it is done through a multi-selection of objects indexes (via a pop-up menu). A function may be defined by entering a mathematical formula or by selecting a function in a set of predefined functions. In some embodiments, the effect defines the function, no additional user parameter is required. In other embodiments, a drop-down list of functions allows the user to select the desired function(s) and obtain the desired behavior.

**[0124]** The parameters displayed on the right each have an effect on the metadata generation function :

- « Parameter amplitude » controls the range of the random pan value. A small amplitude means that the random pan modulation will be small (around 0);
- « parameter offset » allows the control of the center point of the random distribution;
- « Step amplitude » represents the maximum value between two consecutive random values;
- « Speed » represents a manually entered time step at which a new random value is generated;
- « Sharpness » represents whether the object will jump to the new random value or will gradually move towards it;
- « Phase offset » introduces an absolute time delay for the computation of all random values;
- « Phase spread » controls the temporal distribution of the random values for each object of the input group. A phase spread of zero means that all random values are computed at the same time for all objects. A non-zero phase spread introduces a relative time-delay between the computation of random values for each object. The objects positions would then be modulated as a time sequence whose period depends on the number of objects in the group.
- « single source mode » will disregard the stereo properties of input objects, if any.

**[0125]** FIG. 7 shows a flowchart of a method for generating at least one output audio object from a group of input audio objects.

**[0126]** The method may be implemented by a device comprising means for performing one or more or all steps

of the method. The means may include circuitry configured to perform one or more or all steps of the method. The means may include at least one processor and at least one memory storing instructions that, when executed by the at least one processor, cause the device to perform one or more or all steps of a method.

**[0127]** In step 710, a first group of input audio objects is obtained. The input audio objects include respective input audio signals and input object metadata.

**[0128]** In step 720, a user is allowed to define a first function chain including a group analysis function and a metadata generation function.

**[0129]** The group analysis function is configured to perform a statistical and / or comparative analysis of input audio objects in the first group of input audio objects to generate first analysis metadata.

**[0130]** The group analysis function may perform a statistical analysis of the input audio objects in the first group to generate one or more statistical parameters that are part of the first analysis metadata.

**[0131]** The group analysis includes may perform a comparative analysis of metadata of the input audio objects in the first group to generate one or more comparative parameters that are part of the first analysis metadata.

**[0132]** The metadata generation function is configured to generate first output object metadata for at least one first output audio object and has at least one metadata generation parameter (e.g. input parameter or function parameter) computed based on at least one part of the first analysis metadata.

**[0133]** The metadata generation function may be configured to generate the first output object metadata based on at least part of the input object metadata.

**[0134]** The metadata generation function may have at least one metadata generation parameter computed based on time and / or a user-defined temporal event.

**[0135]** A metadata pool that is associated with the function chain may be stored: the metadata pool being configured to store the first analysis metadata generated by the group analysis function. The metadata pool may be configured to store the input object metadata. In one or more embodiments, the user may be allowed to define at least one metadata generation parameter of the metadata generation function based on any of the metadata stored in the metadata pool.

**[0136]** In one or more embodiments, the first function chain includes an audio generation function that is configured to generate audio signal(s). In one or more embodiments, the user may be allowed to define at least one audio generation parameter (e.g. input parameter or function parameter) of the audio generation function based on at least one part of the first analysis metadata.

**[0137]** The audio generation function may be applied by generating at least one output audio signal for output audio object(s) using the audio generation function. The audio generation function may be configured to generate the output audio signal(s) based on one or more input

audio signals. The audio generation function may have at least one parameter computed based on time and / or a user-defined temporal event.

**[0138]** In one or more embodiments, the first function chain includes an audio analysis function configured to perform an analysis of audio signal(s) to generate one or more audio signal metrics. The user may be allowed to define the audio analysis function.

**[0139]** The audio analysis function is applied to generate one or more first audio signal metrics. The one or more first audio signal metrics are stored in the metadata pool for potential use by the audio generation function and / or metadata generation function.

**[0140]** In step 730, the first function chain is applied to the first group of input audio objects. Applying the first function chain may include: generating the first analysis metadata by applying the group analysis function to input audio objects in the first group of input audio objects. Applying the first function chain may include: generating the first output object metadata for the at least one first output audio object using the metadata generation function and the at least one metadata generation parameter computed based on at least one part of the first analysis metadata.

**[0141]** Applying the first function chain may include: generating the one or more first audio signal metrics using the audio analysis function. Applying the first function chain may include: generating output audio signal(s) for output audio object(s) using the audio generation function.

**[0142]** Steps 740-760 may be performed based on the output audio object(s) obtained in step 730 so as to chain the first effect corresponding to the first function chain and a second effect corresponding to a second function chain. Steps 740-760 correspond to the repetition of steps 710-730 but to a second group of input audio objects instead of the first group of audio objects. All embodiments described for steps 710-730 can be applied to steps 740-760 but will not be repeated here for the sake of simplicity.

**[0143]** In step 740, a second group of input audio objects is obtained. The input audio objects include respective second input audio signals and second input object metadata. The second group includes one or more of the output audio object(s) obtained in step 730 and may include other audio objects.

**[0144]** In step 750, a user is allowed to define a second function chain including a second group analysis function and a second metadata generation function.

**[0145]** The second group analysis function is configured to perform a group analysis of second input audio objects in the second group to generate second analysis metadata.

**[0146]** The second metadata generation function is configured to generate second output object metadata for at least one second output audio object. The second metadata generation function has at least one second metadata generation parameter (e.g. input parameter or

function parameter) computed based on at least one part of the second analysis metadata.

**[0147]** In step 760, the second function chain is applied to the second group of input audio objects. Applying the second function chain may include: generating the second analysis metadata by applying the second group analysis function to second input audio objects in the second group of input audio objects. Applying the second function chain may include: generating the second output object metadata for the at least one second output audio object using the second metadata generation function and the at least one second metadata generation parameter computed based on at least one part of the second analysis metadata.

**[0148]** It should be appreciated by those skilled in the art that any functions, engines, block diagrams, flow diagrams, state transition diagrams, flowchart and / or data structures described herein represent conceptual views of illustrative circuitry embodying the principles of the invention. Similarly, it will be appreciated that any flow charts, flow diagrams, state transition diagrams, pseudo code, and the like represent various processes.

**[0149]** Although a flow chart may describe operations as a sequential process, many of the operations may be performed in parallel, concurrently or simultaneously. Also some operations may be omitted, combined or performed in different order. A process may be terminated when its operations are completed but may also have additional steps not disclosed in the figure or description. A process may correspond to a method, function, procedure, subroutine, subprogram, etc. When a process corresponds to a function, its termination may correspond to a return of the function to the calling function or the main function.

**[0150]** Each described unit, function, engine, block, step described herein can be implemented in hardware, software, firmware, middleware, microcode, or any suitable combination thereof.

**[0151]** When implemented in software, firmware, middleware or microcode, instructions to perform the necessary tasks may be stored in a computer readable medium that may be or not included in a device. The instructions may be transmitted over the computer-readable medium and be loaded onto the device. The instructions are configured to cause the device to perform one or more functions disclosed herein. For example, as mentioned above, according to one or more examples, at least one memory may include or store instructions, the at least one memory and the instructions may be configured to, with at least one processor, cause the device to perform the one or more functions. Additionally, the processor, memory and instructions, serve as means for providing or causing execution by the device of one or more functions disclosed herein.

**[0152]** The device may be a general-purpose computer and / or computing system, a special purpose computer and / or computing system, a programmable processing device, a machine, etc. The device may be

or may include or may be part of: a user equipment, client device, mobile phone, laptop, computer, data server, computer, cloud-based server, web server, application server, proxy server, etc.

**[0153]** FIG. 8 illustrates an example embodiment of a device 9000. The processing device 9000 may be used for performing one or more or all steps of any method disclosed herein.

**[0154]** As represented schematically, the device 9000 may include at least one processor 9010 and at least one memory 9020. The device 9000 may include one or more communication interfaces 9040 (e.g. network interfaces for access to a wired / wireless network, including Ethernet interface, WIFI interface, etc) connected to the processor and configured to communicate via wired / non wired communication link(s). The device 9000 may include user interfaces 9030 (e.g. keyboard, mouse, display screen, etc) connected with the processor. The device 9000 may further include one or more media drives 9050 for reading a computer-readable storage medium (e.g. digital storage disc 9060 (CD-ROM, DVD, Blue Ray, etc), USB key 9080, etc). The processor 9010 is connected to each of the other components 9020, 9030, 9040, 9050 in order to control operation thereof.

**[0155]** The memory 9020 may include a random-access memory (RAM), cache memory, non-volatile memory, backup memory (e.g., programmable or flash memories), read-only memory (ROM), a hard disk drive (HDD), a solid-state drive (SSD) or any combination thereof. The ROM of the memory 9020 may be configured to store, amongst other things, an operating system of the device 9000 and / or one or more computer program code of one or more software applications. The RAM of the memory 9020 may be used by the processor 9010 for the temporary storage of data.

**[0156]** The processor 9010 may be configured to store, read, load, execute and/or otherwise process instructions 9070 stored in a computer-readable storage medium 9060, 9080 and / or in the memory 9020 such that, when the instructions are executed by the processor, causes the device 9000 to perform one or more or all steps of a method described herein for the concerned device 9000.

**[0157]** The instructions may correspond to program instructions or computer program code. The instructions may include one or more code segments. A code segment may represent a procedure, function, subprogram, program, routine, subroutine, module, software package, class, or any combination of instructions, data structures or program statements. A code segment may be coupled to another code segment or a hardware circuit by passing and/or receiving information, data, arguments, parameters or memory contents. Information, arguments, parameters, data, etc. may be passed, forwarded, or transmitted via any suitable technique including memory sharing, message passing, token passing, network transmission, etc.

**[0158]** When provided by a processor, the functions

may be provided by a single dedicated processor, by a single shared processor, or by a plurality of individual processors, some of which may be shared. The term "processor" should not be construed to refer exclusively to hardware capable of executing software and may implicitly include one or more processing circuits, whether programmable or not. A processor or likewise a processing circuit may correspond to a digital signal processor (DSP), a network processor, an application specific integrated circuit (ASIC), a field programmable gate array (FPGA), a System-on-Chips (SoC), a Central Processing Unit (CPU), an arithmetic logic unit (ALU), a programmable logic unit (PLU), a processing core, a programmable logic, a microprocessor, a controller, a microcontroller, a microcomputer, a quantum processor, any device capable of responding to and/or executing instructions in a defined manner and/or according to a defined logic. Other hardware, conventional or custom, may also be included. A processor or processing circuit may be configured to execute instructions adapted for causing the device to perform one or more functions disclosed herein for the device.

**[0159]** A computer readable medium or computer readable storage medium may be any tangible storage medium suitable for storing instructions readable by a computer or a processor. A computer readable medium may be more generally any storage medium capable of storing and/or containing and/or carrying instructions and/or data. The computer readable medium may be a non-transitory computer readable medium. The term "non-transitory", as used herein, is a limitation of the medium itself (i.e., tangible, not a signal) as opposed to a limitation on data storage persistency (e.g., RAM vs. ROM).

**[0160]** A computer-readable medium may be a portable or fixed storage medium. A computer readable medium may include one or more storage device like a permanent mass storage device, magnetic storage medium, optical storage medium, digital storage disc (CD-ROM, DVD, Blue Ray, etc), USB key or dongle or peripheral, a memory suitable for storing instructions readable by a computer or a processor.

**[0161]** A memory suitable for storing instructions readable by a computer or a processor may be for example: read only memory (ROM), a permanent mass storage device such as a disk drive, a hard disk drive (HDD), a solid state drive (SSD), a memory card, a core memory, a flash memory, or any combination thereof.

**[0162]** In the present description, the wording "means configured to perform one or more functions" or "means for performing one or more functions" may correspond to one or more functional blocks comprising circuitry that is adapted for performing or configured to perform the concerned function(s). The block may perform itself this function or may cooperate and / or communicate with other one or more blocks to perform this function. The "means" may correspond to or be implemented as "one or more modules", "one or more devices", "one or more

units", etc. A "processing unit" may correspond for example to means for performing one or more processing functions. The means may include at least one processor and at least one memory including computer program code, wherein the at least one memory and the computer program code are configured to, with the at least one processor, cause a device to perform the concerned function(s).

**[0163]** The term circuitry may cover digital signal processor (DSP) hardware, network processor, application specific integrated circuit (ASIC), field programmable gate array (FPGA), etc. The circuitry may be or include, for example, hardware, programmable logic, a programmable processor that executes software or firmware, and/or any combination thereof (e.g. a processor, control unit/entity, controller) to execute instructions or software and control transmission and receptions of signals, and a memory to store data and/or instructions.

**[0164]** Although the terms first, second, etc. may be used herein to describe various elements, these elements should not be limited by these terms. These terms are only used to distinguish one element from another. For example, a first element could be termed a second element, and similarly, a second element could be termed a first element, without departing from the scope of this disclosure. As used herein, the term "and/or," includes any and all combinations of one or more of the associated listed items.

**[0165]** The terminology used herein is for the purpose of describing particular embodiments only and is not intended to be limiting. As used herein, the singular forms "a," "an," and "the," are intended to include the plural forms as well, unless the context clearly indicates otherwise. It will be further understood that the terms "comprises," "comprising," "includes," and/or "including," when used herein, specify the presence of stated features, integers, steps, operations, elements, and/or components, but do not preclude the presence or addition of one or more other features, integers, steps, operations, elements, components, and/or groups thereof.

**[0166]** While aspects of the present disclosure have been particularly shown and described with reference to the embodiments above, it will be understood by those skilled in the art that various additional embodiments may be contemplated by the modification of the disclosed machines, systems and methods without departing from the scope of what is disclosed. Such embodiments should be understood to fall within the scope of the present disclosure as determined based upon the claims and any equivalents thereof.

## Claims

1. A method for generating at least one output audio object, the method comprising:

obtaining (710) a first group of input audio ob-

- jects associated with respective input audio signals and input object metadata;  
 allowing (720) a user to define a first function chain including a group analysis function and a metadata generation function, wherein the group analysis function is configured to perform a statistical and / or comparative analysis of input audio objects in the first group to generate first analysis metadata, wherein the metadata generation function is configured to generate first output object metadata for the at least one output audio object based on at least one metadata generation parameter computed based on at least one part of the first analysis metadata; applying (730) the first function chain to the first group of input audio objects, wherein applying the first function chain includes:
- generating the first analysis metadata by applying the group analysis function to the input audio objects in the first group;  
 generating the first output object metadata for the at least one output audio object using the metadata generation function.
2. The method of claim 1, comprising:
- storing a metadata pool associated with the first function chain, the metadata pool including the first analysis metadata and the input object metadata;  
 allowing the user to define at least one metadata generation parameter of the metadata generation function based on any of the metadata stored in the metadata pool.
3. The method of claim 1 or 2, wherein the first function chain includes an audio generation function configured to generate at least one output audio signal for the at least one output audio object, the method comprising:
- allowing a user to define the audio generation function having at least one audio generation parameter computed based on at least one part of the first analysis metadata;  
 wherein applying the first function chain includes generating the at least one output audio signal using the audio generation function.
4. The method of claim 3, comprising:  
 wherein the metadata generation function is configured to generate the output object metadata based on at least part of the input object metadata.
5. The method of any of claims 2 to 4, comprising:  
 wherein the audio generation function is configured to generate the at least one output audio signal based on at least one of the input audio signals.
6. The method of any of claims 1 to 5, wherein the first function chain includes an audio analysis function configured to perform an analysis of at least one audio signals to generate one or more audio signal metrics, the method comprising:
- allowing a user to define the audio analysis function;  
 wherein applying the first function chain includes generating one or more first audio signal metrics using the audio analysis function for performing an analysis of at least one of the input audio signals, the metadata pool including the one or more first audio signal metrics.
7. The method of any of claims 1 to 6, wherein the first analysis metadata includes a statistical parameter, wherein applying the group analysis function includes performing a statistical analysis of the input audio objects in the first group to generate the statistical parameter.
8. The method of any of claims 1 to 7, wherein the first analysis metadata includes a comparative parameter, wherein applying the group analysis function includes performing a comparative analysis of metadata of the input audio objects in the first group to generate the comparative parameter.
9. The method of any of claims 1 to 8, wherein the metadata generation function has at least one parameter computed based on time.
10. The method of any of claims 1 to 9, wherein the metadata generation function has at least one parameter computed based on a user-defined temporal event.
11. The method of any of claims 3 to 10, wherein the audio generation function has at least one parameter computed based on a user-defined temporal event.
12. The method of any of claims 1 to 11, comprising:
- obtaining a second group of input audio objects associated with respective second input audio signals and second input object metadata, wherein the second group includes the at least one output audio object;  
 allowing a user to define a second function chain including a second group analysis function and a second metadata generation function, wherein the second group analysis function is configured to perform a statistical and / or comparative analysis of input audio objects in the second group to generate second analysis metadata, wherein the metadata generation function is configured

to generate second output object metadata for at least one second output audio object and has at least one second metadata generation parameter computed based on at least one part of the second analysis metadata; 5  
 applying the second function chain to the second group of input audio objects, wherein applying the second function chain includes:

generating the second analysis metadata 10  
 by applying the second group analysis function to the input audio objects in the second group;  
 generating the second output object metadata for at least one second output audio 15  
 object using the second metadata generation function.

13. The method of claim 12, comprising: 20

storing a second metadata pool associated with the second function chain, the metadata pool including the second analysis metadata and the second input object metadata;  
 allowing the user to define at least one metadata 25  
 generation parameter of the metadata generation function based on any of the metadata stored in the second metadata pool.

14. The method of claim 12 or 13, comprising: 30

allowing a user to define a second audio generation function that is configured to generate at least one audio signal and has at least one second audio generation parameter computed 35  
 based on at least one part of the second analysis metadata;  
 generating at least one second output audio signal for the at least one second output audio object using the second audio generation function. 40

15. A device comprising means for performing a method of any of claims 1 to 14.

16. The device according to claim 15, wherein the means 45  
 comprise

- at least one processor;  
 - at least one memory storing instructions that, when executed by the at least one processor, 50  
 cause the device to perform the method.

55

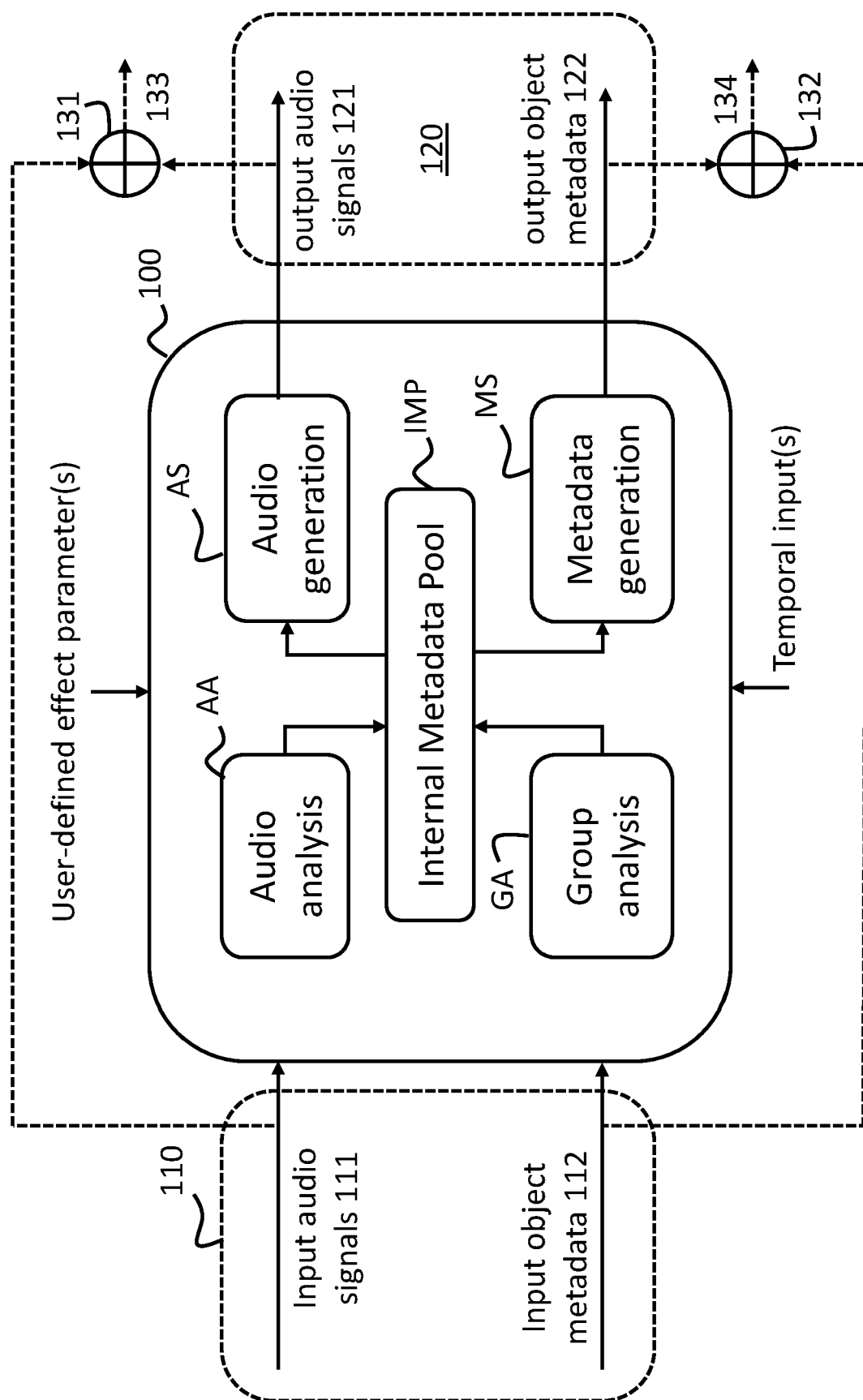


FIG. 1

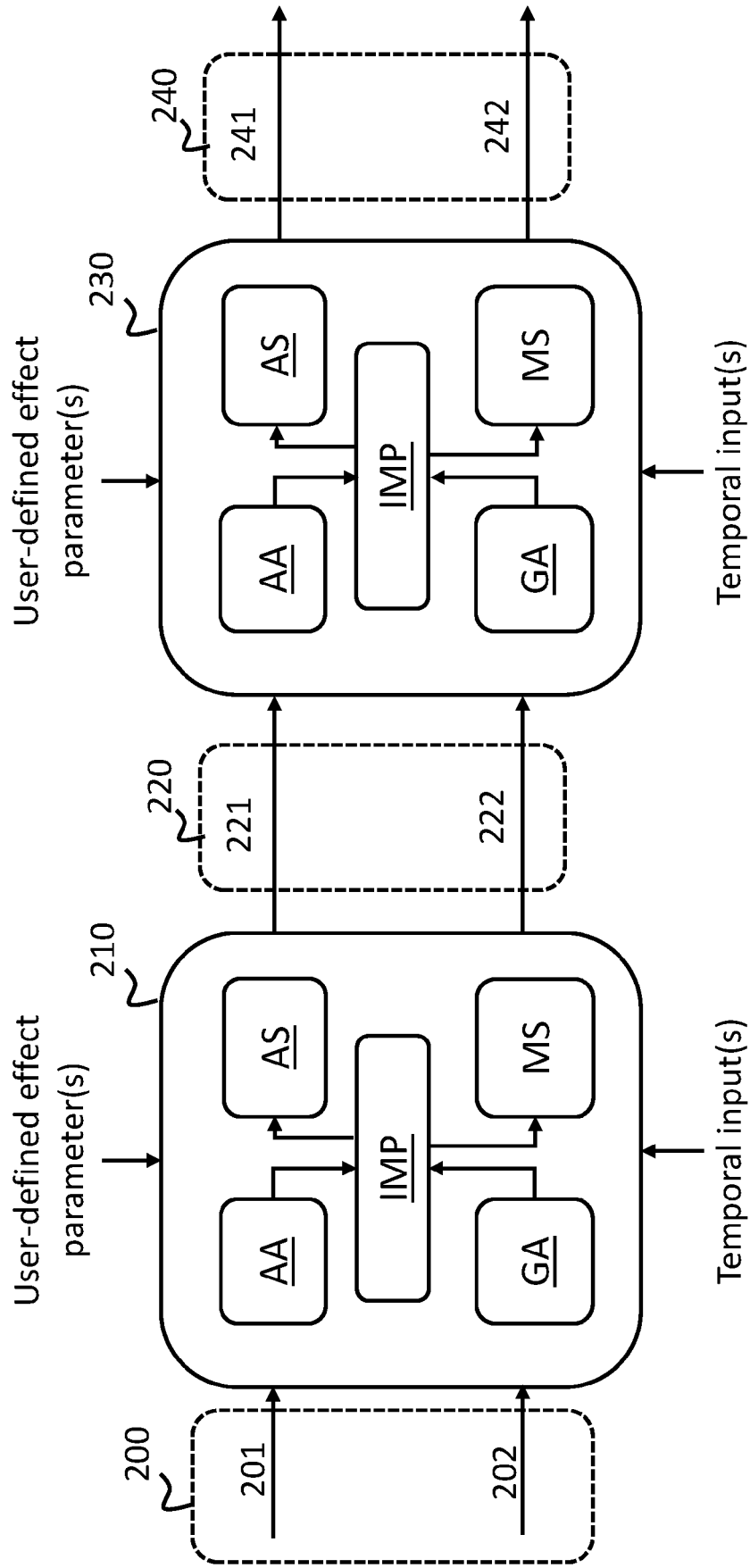


FIG. 2A

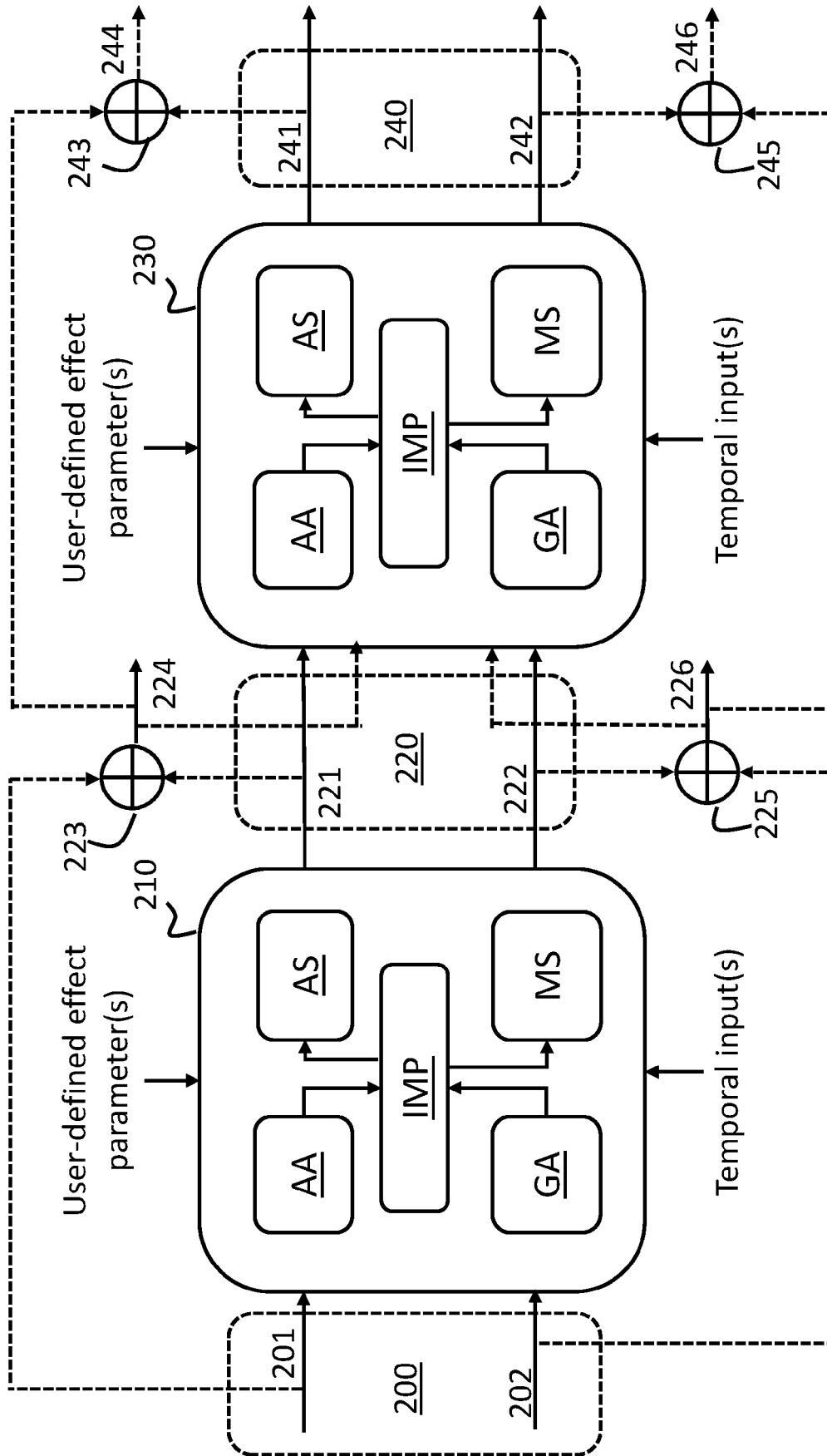


FIG. 2B

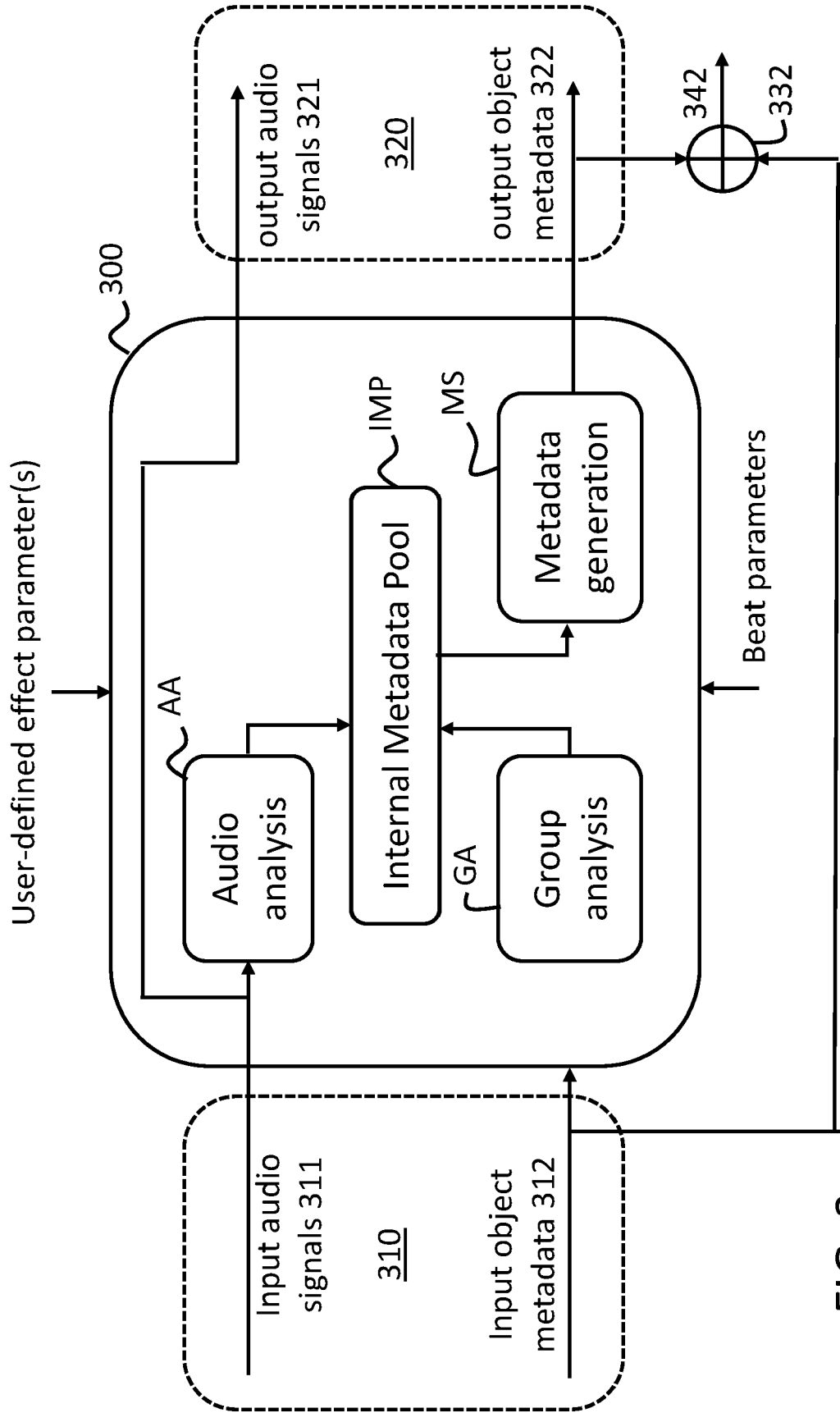


FIG. 3

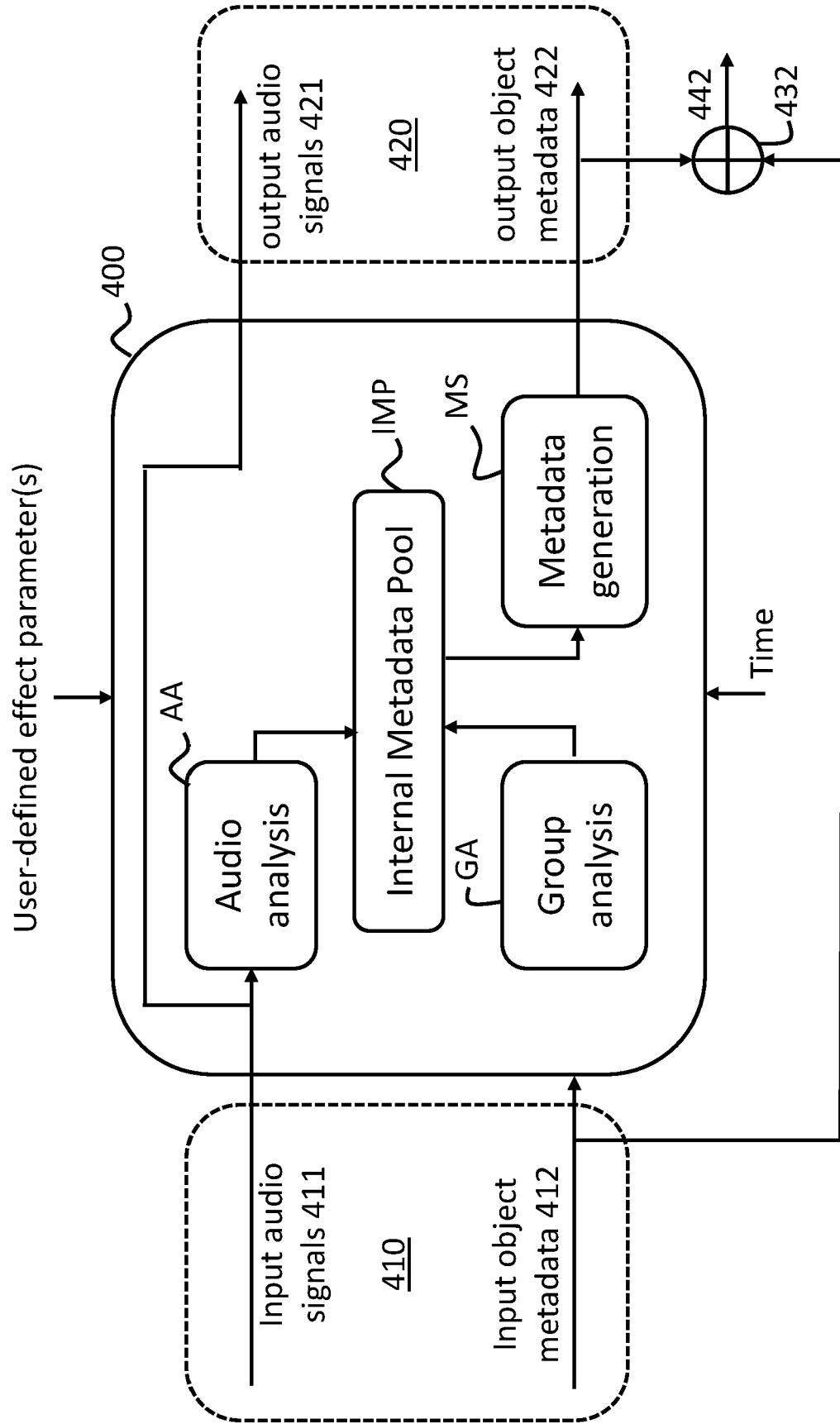


FIG. 4

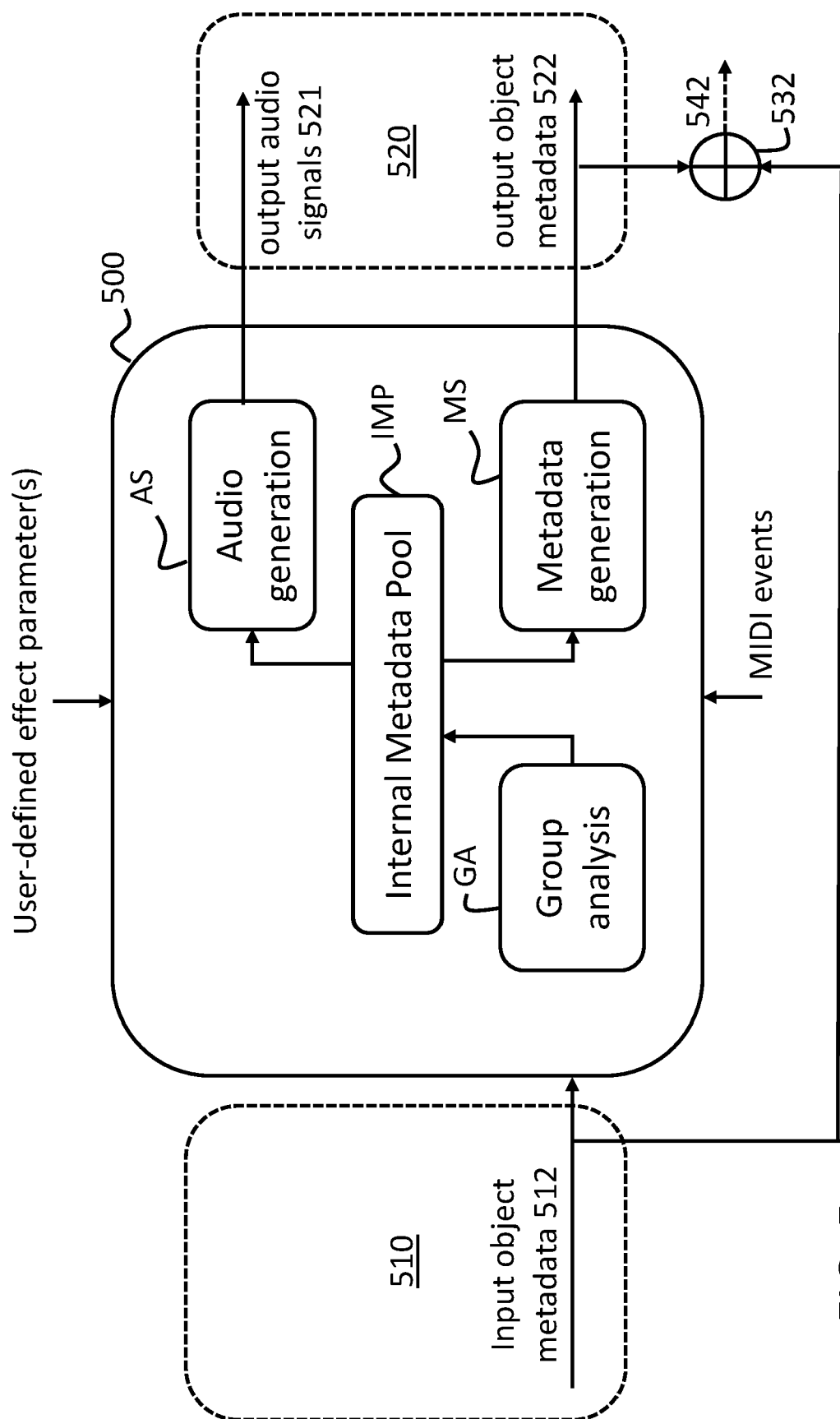
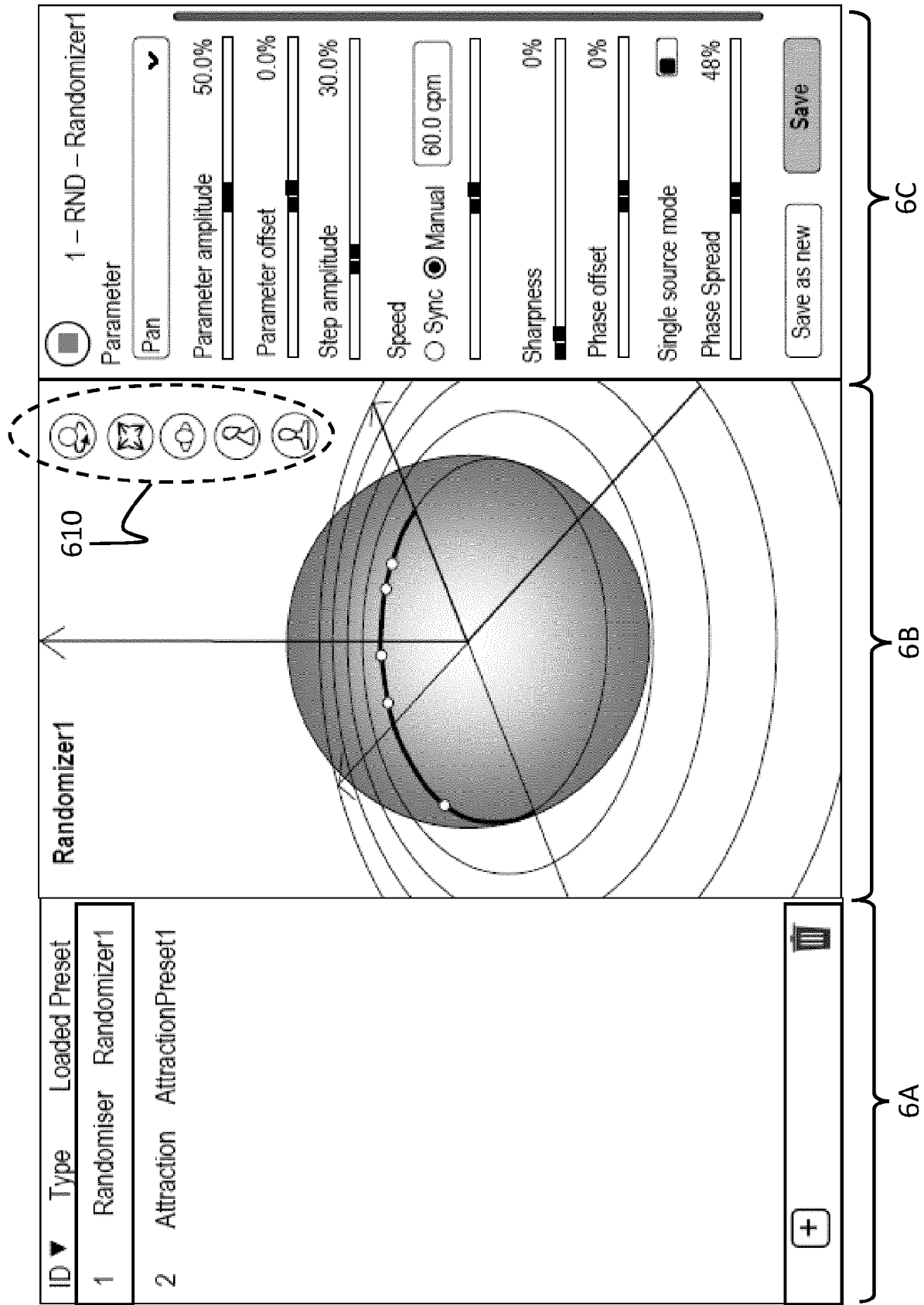


FIG. 5



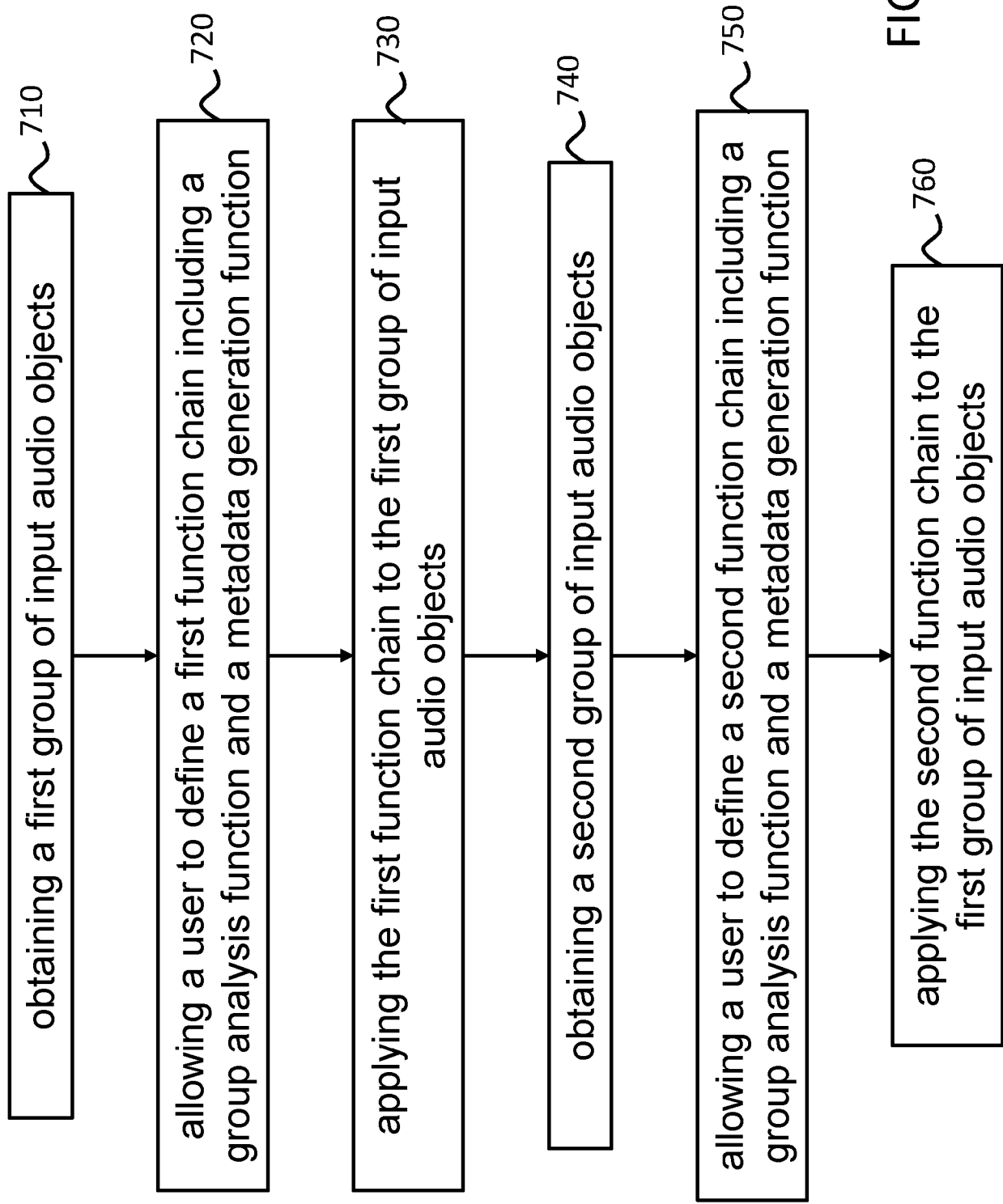


FIG. 7

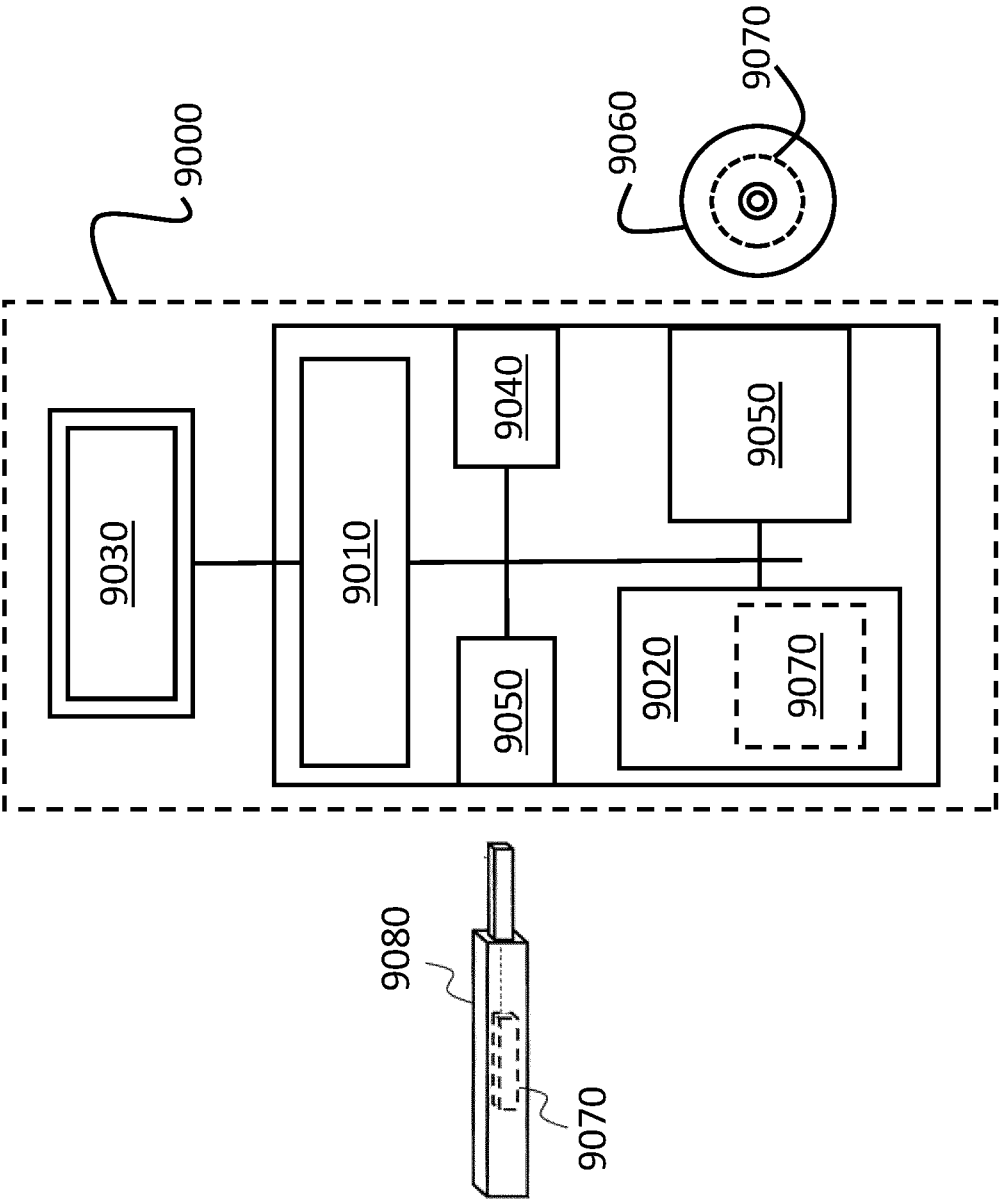


FIG. 8



## EUROPEAN SEARCH REPORT

Application Number

EP 23 30 5234

5

10

15

20

25

30

35

40

45

50

55

## DOCUMENTS CONSIDERED TO BE RELEVANT

Category	Citation of document with indication, where appropriate, of relevant passages	Relevant to claim	CLASSIFICATION OF THE APPLICATION (IPC)
X	EP 0 961 523 B1 (SONY FRANCE SA [FR]) 25 August 2010 (2010-08-25) * paragraphs [0006] - [0036]; figures 1, 2 *	1-16	INV. H04S7/00
X	US 2001/055398 A1 (PACHET FRANCOIS [FR] ET AL) 27 December 2001 (2001-12-27) * paragraphs [0006] - [0010], [0177] - [0198]; figures 1, 2 *	1, 15, 16	
X	Pachet Francois ET AL: "MusicSpace: a Constraint-Based Control System for Music Spatialization", Proceedings of ICMC 1999, 27 October 1999 (1999-10-27), pages 1-4, XP055834465, Retrieved from the Internet: URL:https://www.francoispachet.fr/wp-content/uploads/2021/01/pachet-99-ICMC99.pdf [retrieved on 2021-08-24] * the whole document *	1, 15, 16	
			TECHNICAL FIELDS SEARCHED (IPC)
			H04S

The present search report has been drawn up for all claims

2

Place of search

Date of completion of the search

Examiner

Munich

2 August 2023

Righetti, Marco

## 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

EPO FORM 1503 03/82 (P04C01)

# **ANNEX TO THE EUROPEAN SEARCH REPORT ON EUROPEAN PATENT APPLICATION NO.**

EP 23 30 5234

5 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.

02-08-2023

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
<b>EP 0961523 B1</b>	<b>25-08-2010</b>	<b>EP 0961523 A1</b>	<b>01-12-1999</b>
		<b>JP 2000069600 A</b>	<b>03-03-2000</b>
		<b>US 6826282 B1</b>	<b>30-11-2004</b>
<b>US 2001055398 A1</b>	<b>27-12-2001</b>	<b>EP 1134724 A2</b>	<b>19-09-2001</b>
		<b>JP 4729186 B2</b>	<b>20-07-2011</b>
		<b>JP 2001306081 A</b>	<b>02-11-2001</b>
		<b>US 2001055398 A1</b>	<b>27-12-2001</b>