



(11)

EP 4 682 864 A1

(12)

EUROPEAN PATENT APPLICATION

(43) Date of publication:
21.01.2026 Bulletin 2026/04

(51) International Patent Classification (IPC):
G10H 1/00 (2006.01)

(21) Application number: **24188536.7**

(52) Cooperative Patent Classification (CPC):
G10H 1/0025; G10H 2210/056; G10H 2210/071;
G10H 2210/101; G10H 2210/125; G10H 2240/131;
G10H 2250/311

(22) Date of filing: **15.07.2024**

(84) 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:
GE KH MA MD TN

(72) Inventors:
• **Lyon, Peter**
5227 Veliko Tarnovo (BG)
• **Daya, Nadeem**
London, N10 1BJ (GB)

(74) Representative: **Herbjørnsen, Rut**
Trona Patentrådgivning AB
C/o QuickOffice
Telefonvägen 30
126 26 Stockholm (SE)

(71) Applicant: **Vybez Ltd**
London N10 1BJ (GB)

(54) **ARRANGEMENTS, A METHOD AND A COMPUTER PROGRAM PRODUCT FOR MIXING MUSIC TRACKS**

(57) Music tracks and stems of music tracks may be mixed together by rearranging the order of segments of at least one music track based on a mapping so that their structure over time will match the structure of at least one

other music track. A central database may provide the tracks, stems and analysis data to enable the mapping and rearranging.

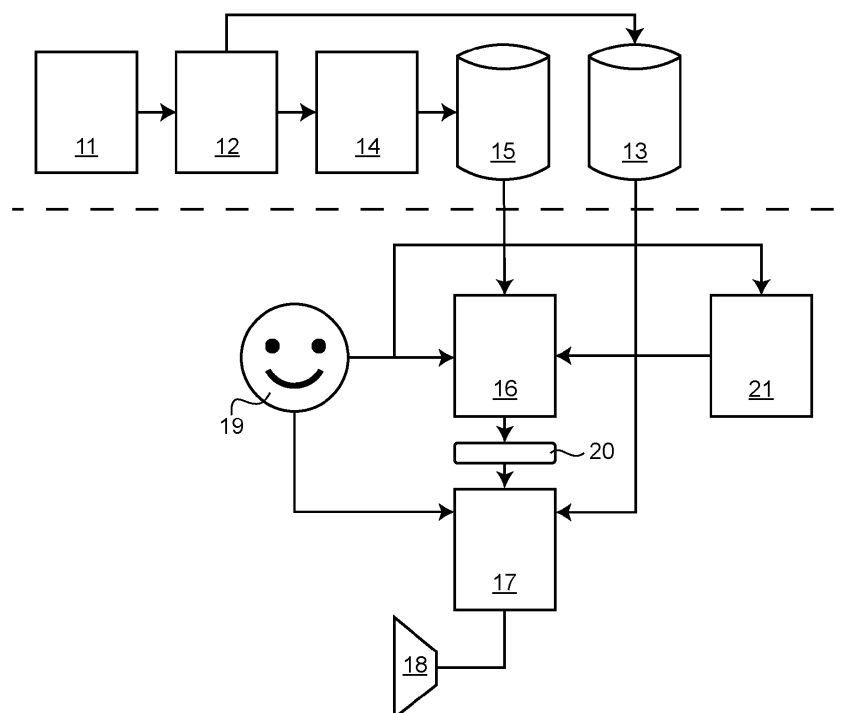


Fig. 1

DescriptionTechnical Field

5 **[0001]** The present invention relates to an arrangement and a method of mixing music tracks, a computer program product for performing the method, and to a player unit for use in such an arrangement.

Background

10 **[0002]** Computer-based systems and methods for mixing music tracks have existed for a long time, with increasingly refined tools and methods. Simply transitioning from one music track to another, including the adjustment of tempo, pitch, etc. is well known in the art. For more flexible combinations, a music track may be separated into different components that may be handled individually.

[0003] In some cases, a music track is presented as a combination of separate components. Alternatively, it is also known in the art to separate music tracks into components, which in this document will be referred to as "stems". Stems can be selected in any suitable way. For example, there may be one stem for vocals and one for each different instrument, for example, drums, bass, guitar and piano. Alternatively, there may be one stem for the rhythm instruments, one for the melody and separate stems for other instruments. A simple implementation would be to have one stem for the melody, such as vocals, and one for everything else. Several different programs exist for generating stems from music tracks.

20 **[0004]** US patent publication No. 2023/0360618 A1 relates to the combination of stems from different music tracks. In particular, this prior art application suggests including a vocal stem from one music track in another music track. The two tracks are matched based, i.a., on beat analysis, where the vocal stem of the first music track is adapted to match the tempo of the second music track and then added to the second music track. US 2023/0360618 does not address the situation where the tempo and character of each track may vary and, therefore, the proposed method will mainly be suitable for limited portions of a music track.

Summary of the invention

30 **[0005]** It is an object of the present disclosure to enable mixes of longer sequences of different music tracks, even if the characteristics of the tracks vary over time.

[0006] The present disclosure therefore relates to an arrangement for creating a music mix based on at least a part of a base music track and an incoming music track, the arrangement comprising a stem remap unit, adapted to:

- obtain a base analysis of the base music track, the base analysis comprising a number of descriptors with respect to its characteristics over time,
- obtain an incoming track analysis of at least one stem of the incoming music track, the incoming track analysis comprising a number of descriptors with respect to the characteristics of at least a first and a second plurality of granules of one stem of the incoming track over time,
- based on a comparison of the descriptors of the base music track and the at least one stem of the incoming music track, rearrange the order of a plurality of granules of the at least one stem to match the characteristics of the base music tracks over time, the rearranging resulting in a first mapping of the stem to the base music track.

[0007] The arrangement further comprises a player unit configured to obtain at least the incoming music track from an original track database and render a first output track to be played based on the at least one stem, and the mapping.

45 **[0008]** The disclosure also relates to a computer-based method of mixing at least parts of a base music track and an incoming music track, comprising

- obtaining, from an analysis database, an analysis of the base music track comprising a number of descriptors with respect to its characteristics over time
- obtaining, from the analysis database an analysis of at least one stem of the incoming music track, comprising a number of descriptors with respect to its characteristics over time and a division of the at least one stem of the incoming music track into a number of granules,
- based on a comparison of the analyses of the base music track and the at least one stem of the incoming music track, rearranging the granules of the at least one stem to match the characteristics of the base music track over time, the rearranging resulting in a first mapping of at least some granules of the stem,
- obtaining at least the base music track and the incoming music track, the incoming music track being split into at least two stems representing different components of the respective music track, and
- rendering an output track to be played, based on, the at least one stem of the incoming music track, and the first

mapping of the at least one segment of the incoming track.

[0009] The arrangement and method outlined above enable the adaptation of a stem of an incoming music track to the structure of a base music track by moving parts of the stem, referred to as granules, around to match the characteristics of the base music track.

[0010] The steps of the method above may be repeated based on a comparison of the analyses of the base music track and the at least one stem of the incoming music track, including rearranging the granules of the at least one stem to match the characteristics of the base music track over time, the rearranging resulting in a second mapping of at least some granules of the stem which is different from the first mapping, obtaining at least the base music track and the incoming music track, the incoming music track being split into at least two stems representing different components of the respective music track rendering an output track to be played, based on at least one stem of the base music track, the at least one incoming track, and the second mapping of the at least one segment of the incoming track. In this way, several alternative versions of a mix based on the same base track can be created. The step of rendering the output track to be played may then involve applying the first mapping and the second mapping at different times during playback of the output track. This enables switching back and forth between different versions during playback.

[0011] Preferably, the output track is rendered based on the at least one stem of the incoming music track, and the first mapping of the at least one granule or sequence of granules of the incoming track and at least one stem of the base music track. In this case, the player unit is further configured to obtain the base music track, and render the first output track based on the at least one stem, the mapping and the base music track. This will enable the mix of the at least one stem and the base track to be played together.

[0012] According to preferred embodiments, the granules are rearranged so that a number of consecutive granules form a segment and the mapping involves rearranging the order of at least a first and a second segment. The method then suitably involves creating at least a first and a second segment, each segment comprising a plurality of consecutive granules, wherein the step of rearranging the granules involves rearranging the at least first and second segment. The grouping of granules into segments enable a more efficient handling of the different parts of the incoming stem. Preferably, the segments can vary in size so that consecutive granules having similar properties, or matching a particular portion of the base track can be handled as one unit.

[0013] The arrangement may further comprise a mutation generator arranged to generate at least a first mutation of the first mapping to create a new mapping of the stem to the base music track, and a selection unit arranged to select the first mapping or the new mapping.

[0014] Optionally, the arrangement may further comprise a suggestion engine arranged to obtain a base analysis of the base music track, the base analysis comprising a number of descriptors with respect to its characteristics over time, select an alternative incoming music track and obtain an alternative incoming track analysis of at least one stem of the alternative incoming music track, the incoming track analysis comprising a number of descriptors with respect to the characteristics of at least a first and a second plurality of granules of one stem of the alternative incoming music track over time, and generate a suggested mapping based on the base music track analysis and the alternative incoming track analysis. The player unit is the preferably arranged to render the output track as a combination of the first mapping and the suggested mapping or several suggested mappings.

[0015] The disclosure also relates to an arrangement for providing music tracks suitable for mixing, comprising:

- a mastering and stem splitting unit arranged to receive a first music track and split it into at least a first and a second stem, creating an original track,
- an analysis unit arranged to derive descriptors from the original track and/or stems, the descriptors relating to how the qualities of the original track and/or stem change over time,
- an analysis database arranged to store the result of the analysis,
- an original track database arranged to store the first music track in the form of stems, in a format that may be used to create a playable track.

[0016] This arrangement will provide the information used by the above-described arrangement for creating a music mix. The analyses are preferably performed off-line and the arrangement may serve a number of users creating music mixes, providing them with tracks, stems and analysis data.

[0017] The disclosure also relates to a player for use in an arrangement for creating a music mix according to the above, said player being arranged to obtain a mapping of at least a first incoming stem, said mapping representing an order of segments of the at least first incoming stem, obtain the first incoming stem, render a reordered version of the stem based on the mapping divide the stem into chunks to facilitate playback of the stem according to the mapping, optionally together with a base track.

[0018] The player may be configured to instantly switch from a first to a second version of audio based on the same base track, by determining a current beat position in the base track, and switching from a first timecode in the first version to a

second timecode in the second version, both the first and the second timecode corresponding to the current beat position in the base track, to maintain musical continuity.

[0019] Hence, the disclosure relates to an arrangement and a method for dynamically creating new musical pieces according to user requests by combining music from multiple donor pieces. Donor pieces are split up into their donor 'parts' (e.g. drums, bass, instruments, vocals) to create donor stems. Multiple analysis descriptors are derived from each donor piece, and a Machine Learning (ML) implementation is used to explore to find good linear and nonlinear time-mutations that can be applied to one or more of the donor stems, that will ensure any structural and compositional conflicts that may be created by combining two parts from two different donor pieces. The resulting new musical piece is then rendered and played to the user by applying the calculated mutations to the donor parts and optionally saved in PCM audio format to be exported for external sharing and/or use in other applications.

[0020] The disclosure also relates to a computer program product comprising computer readable code means which, when executed in a computer, will cause the computer to perform the method according to any embodiment of the disclosure. The disclosure also relates to a non-transitory storage unit having stored thereon computer readable code means which, when executed in a computer, will cause the computer to perform the method according to any embodiment the disclosure.

Terminology

[0021] The following terminology will be used in this document:

- Raw track - a piece of music, as delivered by a rightsholder. A raw track may be used as input to the mixing processes discussed in this document. Raw tracks are supplied in either single-track masters or multi-stem projects already separated into stems.
- Original track - a raw track that has been processed and divided into stems.
- Stem - a musical part from a raw track, for example the vocal part or a particular instrument. The stem lasts for the entire duration of the track.
- Remix - A combination of stems and/or structure derived from two or more original tracks. A remix has its own key and BPM.
- Base track - The track on which the musical structure of a remix is based.
- Incoming track: A track from which one or more stems may be taken and added to the structure extracted from the base track to create a new version of the base track.
- Remaster - the act of making changes to the musical characteristics of a track, including changing the speed, pitch, balance between stems and the addition of sound effects to the track.
- Granule - The precision of the analysis and mapping description, defined in terms of beats. For many applications a suitable granule size is one beat, but it may also be longer or shorter than one beat.
- Segment - A time-slice from a stem, comprising a number of consecutive granules. The number of granules in each segment may vary, so that the segments may differ in size.
- Chunk - A time-slice from a stem, for the purpose of decoding and buffering during playback. Chunks normally have the same length.
- Mapping - a description of how to re-arrange the flow of a stem, described in terms of segments and including independent pitch and playback rate shifting. Thus, a mapping is a data structure that describes a solution to resolving the structural and harmonic conflicts between a stem that is taken from an incoming track and the base track.
- Version - a remix derived from a base track. Multiple remixes derived from the same base track will be referred to as versions of the remix.
- Playable track - Either an original track or a remix.

[0022] Further, the term instantly in this document is taken to mean that no latency is induced by the software system over the latency of the native device and/or runtime environment. The term realtime does not necessarily mean instantly, but with a short enough delay to maintain user engagement, for example a response to a user request within a maximum of 3 seconds.

Brief description of drawings

[0023] The invention will be described in more detail in the following, by way of examples and with reference to the appended drawings.

Fig. 1 is an overview of a system that may be used to implement embodiments of the invention.

Fig. 2 illustrates the division of a stem into segments and the rearrangement of segments.

Fig. 3 is a flow chart of a possible method according to a first aspect of the disclosure.

Fig. 4 is a flow chart of a possible method according to a second aspect of the disclosure.

Figs. 5 - 7 are flow charts of how to implement different types of analyses

Fig. 8 illustrates the function of Tabu Search as an example of an unsupervised optimization technique.

Fig. 9 illustrates the function of the fitness evaluator shown in Fig. 8.

Detailed description of embodiments

[0024] There are a number of key objectives that influence the design and software architecture of the system. Ideally, the system should be portable, meaning that the system should be capable of running on all modern devices, available via the browser, or installed through a native application. Further, the system should be able to create any remix within a short period of time, for example 5 seconds, of the user requesting it. Once a remix has been created users should be able to switch back and forth instantly between versions of a track, respecting the musical flow.

[0025] The memory usage of the audio generation components should be limited so as to ensure that the application can be run in the background and across a range of devices without need for special consideration by the user. In practice, this currently means a limit of approximately 500MB, although this will depend on the devices used and on future technological development.

[0026] The user should also have the ability to instantly remaster any track at any point in time while listening to it and should be able to combine stems from any track in our library in real time to allow them to flexibly create new sounding remixes.

[0027] For the purpose of allocating IP ownership rights to the remix in dependence of the base and incoming tracks, the system should produce a roster for each new Remix describing the exact composition.

[0028] The system should be capable of rendering each user-created remix entirely on an end user device, to ensure that the responsiveness and performance of the system is not dependent on server-side processing, and does not experience issues with high-volume scaling. The server requirements should only include distribution of audio files of original tracks, and the creation of new remixes should not require additional storage space apart from the list of required stems and calculated mappings for the remix.

[0029] Optionally, the system may be capable of rendering and exporting the remix in PCM, MP3 or other common audio formats, either on a client device or server for the purpose of external sharing, use in other applications or any other use where distribution of the rendered track is more convenient or efficient.

[0030] Fig. 1 is an overview of a system that may be used according to the present invention. The units above the dashed line perform the centralized processing, which may be done offline, to provide music tracks and other information that may later be used to create remixes. These units include a mastering and stem-splitting unit 12, an original track database 13, an offline analysis unit 14 and an analysis database 15. The units involved in the creation of remixes are shown below the dashed line and should work in real-time or near real-time. These units include a stem remap unit 16, a player unit 17, and an output unit 18, represented in Fig. 1 as a loudspeaker, for presenting the music remix to a user 19.

[0031] At the beginning of the centralized processing, a raw track 11 is presented to the mastering and stem-splitting unit 12. If the raw track is provided as a single track, the mastering and stem-splitting unit splits the track into stems. Then all stems undergo a mastering step. The purpose of the mastering step is discussed in more detail below. The output from the mastering and stem-splitting unit 12 is an original track, which is forwarded to the original track database 13 from where it can be accessed by a user wishing to create a remix. The original track is accessible both as the complete track and as the stems of the track. The original tracks may be stored in the original tracks database 13 in any suitable music format, for example MP3.

[0032] The original track is also forwarded to an offline analysis unit 14. The offline analysis unit analyses the original track and creates a number of descriptors based on the original track. The description of each original track is time-based and describes how the qualities of each track change over time. There may be one set of descriptors for the complete track

and one for each stem. Alternatively, the descriptors for the complete track may also be applied to the stems of the track. The analyses may include, but are not limited to: intensity analysis level, harmonic analysis using self similarity matrices (SSM), and musical part analysis. Examples of how to implement harmonic analysis, intensity analysis and musical part analysis are shown in Figs. 5, 6 and 7, respectively.

[0033] The descriptors are forwarded from the analysis unit 14 to an analysis database 15 where they are stored, organized by track, and may be accessed by a user 19 when creating a remix.

[0034] When creating a remix, or remapping, the user 19 sends a new track request to a stem remap unit 16. The request includes a first original track to be used as a base track and a second original track to be used as an incoming track, or one or more stems of the second original track. The remap unit 16 obtains the descriptors regarding each stem and original track included in the request from the analysis database 15 and uses a combination of statistical classification and machine learning techniques to produce a set of mappings 20. This is preferably done in real-time or near real-time. The mapping is a data structure describing a solution to resolving any structural and harmonic conflicts between the base track and stems that may be added from an incoming track. The function of the stem remap unit 16 and the nature of the mapping will be discussed in more detail below.

[0035] When the mapping has been generated, it is provided to the player unit 17. The player unit also obtains the original tracks and/or stems to be used in the remapping from the original track database 13 and renders the playable track for the user by applying the mapping to the original tracks and/or stems. The playable track is then played back using music output equipment 18. The music output equipment may be any kind of computer, mobile device, tablet, etc. capable of playing music tracks. If the remix is being rendered for the purpose of export, then the output unit 18 may be embodied by a PCM audio buffer and optionally an encoder for the required output format, for example, MP3.

[0036] Optionally, the system also comprises a suggestion engine 21. The suggestion engine is arranged to receive new track requests from the user and request the creation of mappings or alternative versions using the same base track as requested by the user. The suggestion engine 21 is arranged to provide suggestions of possible alternative versions that are presented to the user using the graphical user interface, such that while listening to a current version, the user can select and instantly switch between the current version and the possible alternative versions suggested. Typically, the suggestions might include one or more of the stems currently being listened to, as well as a number of alternative stems from other original tracks. If the suggestion engine has provided alternative versions with the same base track, they will be provided to the player so the user can listen to them and further modify and remaster them, as desired. The user can also create multiple versions and switch between them by selecting the incoming tracks or stems manually. The player can preload a number of these versions such that they can be selected instantly when requested by the user. The player 17 may switch from the currently playing version to a new version. It will do so by switching to the same beat position in the base track, to maintain musical continuity during the transition between the two versions.

[0037] The offline analysis unit 14 is arranged to perform a number of algorithms which are all fundamentally based around initially running FFTs of stem data and applying a range of different transformations of that data that result in a set of compactly described bar-by-bar time-series descriptions of the music. This includes a mix of 1-dimensional descriptions e.g. magnitude of signal, group allocations (e.g. chorus, verse, intro), transient timing analysis, and 2-dimensional descriptions like self similarity matrices, which will be discussed in more detail below.

[0038] Once analyzed, the remap unit 16 explores available transform options to find mapping solutions that perform well with as few 'mutations' as possible, a mutation being a step required to map one track into another. This may be done using a machine-learning optimization algorithm to compute the minimum number of mutations to the track flow in order to achieve the best alignment between the analysis criteria. There are suitable machine-learning algorithms for this step, which do not require training. One such algorithm is Tabu Search, which is an unsupervised optimization technique that will optimize the solution within the local space created by the parameters, in this case the base analysis and the incoming analysis. An overview of how this may be achieved is shown in Fig. 8 below.

[0039] With multiple analysis descriptors available for each stem the remap unit 16 uses an ML implementation which tests out potential solutions by applying the mapping transform to each analysis and creating new simulated analyses for the application of the current mapping to the analysis data of the incoming track in question. These analyses can then be evaluated by their similarity to the analyses of the stems from the base track to calculate the difference between the proposed mapped stems and the original stems in all the analysis dimensions available.

[0040] After exerting a predefined level of effort, the remap unit 16 returns the best performing mapping. One way of determining the effort level involves determining when a certain number of iterations of the optimization process have been performed without yielding any significant improvement in performance of the mapping, for example, the point of diminishing returns.

[0041] Fig. 2 illustrates an example of the division into segments, the rearranging of segments performed to be able to use the stems of the incoming track to replace stems of the base track in a meaningful way and the expansion of segments to enable fade-overs between segments during playback.

[0042] Fig. 2 illustrates in a) an original stem, with two segments B and A identified for remapping to create a suitable remapped stem. The segments comprise audio data from the incoming track in the form of a number of consecutive

granules. The number of granules in a segment is preferably variable. Although only two different segments have been identified in this example, it will be understood that any suitable number of segments may be identified, in any suitable parts of the original stem. In b) a remapped stem is shown. For simplicity, in this example, the remapped stem has the segments A and B alternating.

[0043] In c), the segments labelled A have been expanded by adding a bit of time at the beginning and the end of each segment and in d), the segments labelled B have been expanded in the same way, as indicated by grey areas at the beginning and the end of each segment, where they fade in or out, respectively. Accordingly, no extra time has to be added at the beginning of the first segment or the end of the last segment. During playback, when switching from a current segment to a next segment, the volume of the current segment is gradually decreased so that the current segment fades out. At the same time, the next segment is gradually increased from the start of the added bit at the beginning for a suitable period of time. When it has been ramped up to its full volume, it becomes the current segment. In this way, a fadeover is created between the current segment and the next segment.

[0044] The ramping up and ramping down, respectively, of each segment is illustrated in curves c) and d) for segments A and B, respectively. The overlap in time is shown as grey areas at the beginning and the end of each segment.

[0045] The resulting remapped stem is shown as e), with the fade-over periods marked as grey areas around the boundaries between the segments.

[0046] In f), the remapped stem is shown, divided into chunks for the purpose of streaming from the remap unit to the player unit. Chunk boundaries, shown as solid vertical lines in f), are calculated as $\text{last_chunk_boundary} + \text{max_chunk_length}$. If a calculated chunk-end boundary conflicts with the segment fadeover region, such as the boundary indicated by a dashed vertical line 23, then that chunk boundary is brought back to before the start of the conflicting fadeover region as shown by the solid vertical line 24 just before the fadeover region containing the dashed vertical line 23 (the chunk in question is shortened). Alternatively, it may be delayed to the end of the conflicting fadeover region (extending the chunk in question). This avoids having any chunk boundaries during fadeovers, which reduces the complexity of calculations required when constructing chunks. The chunk size may be chosen arbitrarily but is preferably selected as a trade-off between the amount of memory available to run the application and the processor overhead caused by messaging related to the playback. In practice, a suitable value has been found to be around 5 seconds for use on many browsers, but the chunks may be smaller or larger depending on memory requirements.

[0047] The mapping data format describes the transform necessary to be able to use the stems of the incoming track to replace stems of the base track in a meaningful way. The mapping defines a number of segments which are audio data taken from the timeline of the incoming track. Preferably, the segments are described in terms of beats, rather than seconds so as to be agnostic to Stem BPM. These segments are then ordered along a timeline to describe the new timeline of the rearranged stem.

[0048] Optionally, not shown in Fig. 2, one or more periods of silence may be injected in the timeline of the track. Each segment may also have unique effects applied, such as EQ, distortion, pitch shift and time-stretch.

[0049] The player unit 17 in this embodiment is a stream-decoding multi-stem audio player that is capable of performing mapping transformations independently to each track of the audio in realtime while the files are being decoded and played. The player also supports remaster instructions that are applied instantly to the stems of the current track in response to user request. The player unit 17 also supports simultaneously stream decoding multiple versions of a base track (which may be at different BPMs), such that the user can instantly switch back and forth between versions, while maintaining the current beat position in the base track.

[0050] The player unit 17 also supports the loading and decoding of the start point of a second track while the current track is being played such that the user can request the second track and the player should respond instantly.

[0051] Each transition between segments is preferably played back with a short fade-over effect to ensure that there are no audio distortions created as illustrated by Fig. 2. The inclusion of the fade-over effect means that each segment must be extended by $(\text{seconds_fadeover} \div 2)\text{s}$ at the beginning and end of the segment and have a volume ramp applied at the beginning and end of the extended segment from 0 to Max volume lasting seconds fadeover. The fadeover regions that occur where two segments meet (shown in grey) contain data from both segments which are overlapping as a result of being extended.

[0052] Due to the high memory requirements for a simple multi-stem, multi-version audio engine, streaming decoding of the audio is preferably performed by splitting the stem up into chunks, and only loading the necessary decoded chunks into the player at any one time.

[0053] Fig. 3 is a flowchart of a method according to embodiments of one aspect of the invention, for providing a set of original tracks that may be used in remapping. These steps may be performed by the units 11- 15 above the dashed line in Fig. 1. The analysis outlined in Fig. 3 may be performed offline and contains a number of algorithms which are all fundamentally based around initially running Fast Fourier Transforms (FFTs) of stem data and applying a range of different transformations of that data that result in a range of compactly described granule-by-granule time-series descriptions of the music. These types of analyses will be described in more detail below.

[0054] In a first step S31, a raw track is provided. If the raw track is a single file, that is, not already split into stems, it is split

into stems in a following step S32. Any suitable software could be used to split the tracks. After splitting, the stems are optionally mastered in a step S33 to ensure that they have the right quality. This typically includes ensuring that the instruments are grouped appropriately and normalizing the quality and volume of each stem to the existing original tracks in the library if needed. The mastering may be performed by manual effort or existing AI tools, or a combination of the two.

[0055] In an analysis step S34, the stems of the original track are analyzed and labelled with respect to key and beat information, such as bpm and beat grid offset. The analysis step creates a number of descriptors derived from each original track and/or component stem. These descriptors are time-based and describe how the qualities of each track change over time. These analyses include but are not limited to: volume level, harmonic distribution, self-similarity matrices (SSM), and novelty detection, that is, timecodes where dynamics change significantly. As shown in Fig. 1, the result from the analyses is stored in an analysis database, organized by track and stem. The content of the original track database 13 and the analysis database 15 are made available to a user who wishes to create a new remix based on two or more original tracks.

[0056] Fig. 4 is a flowchart of a method according to another aspect of the invention, which may be performed by the units below the dashed line in Fig. 1. This method is performed by a user 19 wishing to create a new remix. In a first step S41, the user submits a new version request to the remap unit 16. The version request typically includes a genre, style or "vibe" for the version, and may also include a specific incoming track name for the mashup. The stem remap unit 16 then, in step S42, obtains from the analysis database 15 the data relating to each stem and original track included in the request and uses a combination of statistical classification and machine learning techniques to produce a set of mappings in real-time. This will be discussed in more detail in connection with Fig. 8.

[0057] In a subsequent step S43, the player receives the mappings from the remap unit 16 and fetches the music data for the new track from the original track database 13, and in step S44 renders the track for the user. The rendering may include the application of filters and mastering controls, which must react instantly to requests from the user. The filters and mastering controls may typically include playback rate, pitch, distortion, and balance between instruments. The filters and mastering controls can optionally be set automatically according to the dynamics extracted from the mapping/analysis process.

[0058] Optionally, if a suggestion engine is available, the suggestion engine 21 may, in a step S45, receive new track requests and request the creation of mappings or alternative versions provide alternative versions using the same base track that was requested by the user 19. The suggestion engine looks at the current state of the player and offers the user statistically determined version options via the user interface. For example, if the user 19 had created an electronic music version of a rock track, the suggestion engine may offer to the user alternative electronic music options to help them find the sound they are looking for.

[0059] If several versions of the same track have been provided, by manual input from the user 19, and/or from the suggestion engine 21, the user may switch between different versions. Preferably, the player unit 17 will then identify the same beat position in the base track in each of the versions to maintain musical continuity during the transition from one version to another. This is advantageous because it accounts for different versions based on the same base track having different BPMs and allows versions to be switched back and forth on the fly without breaking the musical structure that has been defined by the base track. As an example, consider two versions, version A at 60BPM and version B at 120BPM both based on the same base track. After listening to the first 10 seconds of version A, the user selects version B. The player switches from playing timestamp 00: 10 of version A to timestamp 00:05 of version B and plays version B from then onwards. After another 10 seconds, the user switches back to version A. The player switches from playing timestamp 00: 15 of version B to 00:30 of version A.

[0060] The stem remap unit 16 uses machine learning algorithms to explore available transform options and find mapping solutions that perform well with as few 'mutations' (steps required to map one track into another) as possible.

[0061] Fig. 5 is a flow chart of a possible implementation of harmonic analysis of a track. A first main step S51 involves performing a chroma analysis of each stem of the track. This first main step includes the substeps of S511: breaking up the track into small time units which will be referred to as granules, each granule being, for example, one beat, S512: downsampling the frequency space to harmonically relevant information (for example, 0 - >5 kHz), S513: running a Fast Fourier Transform (FFT), S514: sampling FFT frequency data using a windowing function and folding octaves down to describe harmonic distribution as a single octave S515: determining average harmonic distribution across each beat producing a granule description of harmonic distribution of each granule, and in S516 all granules are tied together to describe each stem as a series of 12 value harmonic descriptors (granules), one for each semitone in an octave.

[0062] A second main step S52 involves computing a self-similarity matrix of the granules. This second main step includes S521: computing a matrix of size num_granules * num_granules where the value of each cell is the distance between the row and column granule descriptors, and S522: producing a harmonic self-similarity matrix by normalizing and inverting the granule matrix such that granules with the lowest distance between them have a value of 1.0 and the furthest away granules have a value of 0.0.

[0063] With reference to the second main step above, an example of the substeps S521 - S522 is shown below. Assuming that the granules are as shown in Table 1:

Table 1, example of granules

Granule index	Granule Descriptor
0	[0,0,0,0.5,0,0,0,0,1,0,0]
1	[0,1,0,0,0,0,0,0,0,0,0]
2	[0,0,0,0.5,0,0,0,0,1,0,0]
3	[0,1,0,0,0,0,0,0,0,0,0]

The computed matrix will be as shown in Table 2:

Granules	0	1	2	3
0	0	2.5	0	2.5
1	2.5	0	2.5	0
2	0	2.5	0	2.5
3	2.5	0	2.5	0
Table 2, a computed self-similarity matrix for the granules of Table 1				

The normalized and inverted matrix according to step b) will be as shown in Table 3:

Granules	0	1	2	3
0	1	0	1	0
1	0	1	0	1
2	1	0	1	0
3	0	1	0	1
Table 3, the harmonic self-similarity matrix based on the matrix of table 2				

[0064] Fig. 6 shows an example of an intensity analysis implementation. In a first step S61, the stems are split into granules, for example one beat each. In a second step S63, the intensity analysis of each granule is performed and in a third step S63, all the granules are tied together and normalized to describe each stem as a time series of 1-value intensity descriptors. This is achieved by performing a low-resolution FFT across the granule, and taking the sum of the real components of the FFT as the overall amplitude.

[0065] Table 4 shows an example of a stem divided into four granules described in terms of intensity.

Granule	Value
0	12
1	14
2	18
3	3
Table 4	

The intensity descriptor for each stem may be normalized by setting the highest value to 1. For the example shown in Table 4, this yields the normalized matrix of Table 5:

Granule	Value
0	0.667
1	0.778

(continued)

Granule	Value
2	1.000
3	0.167
Table 5	

[0066] Fig. 7 shows an example implementation of a musical part analysis. This analysis system results in a description of the track in terms of, for example, "parts" and granule ranges for each part. The parts may be intro, verse, chorus, bridge, etc. Table 6 shows an example of what this may look like for an example track:

Part	Start Granule	End Granule
Intro	0	24
Verse	24	48
Chorus	48	72
Verse	72	96
Outro	96	108
Table 6. identification of parts of an example music track and the granules involved in each		

[0067] For the purpose of describing the track, a convolutional neural network trained on available music data sets including tracks and for each track an identification of the parts and the granule range for each part may be used. In a first step S71, the track is treated as a whole track, with all stems merged together into one master track. In a subsequent step S72, the master track is divided into granules and chroma analysis is performed, as described in connection with Fig. 5. Next, an image representation of each track is created. This may suitably be a 12*num_granules pixels size image. This creates a pseudo-spectrogram that is in terms of semitones rather than pure frequency (12 values that include the frequency data from every octave). The spectrogram has the same precision as the size of the granule, so the resulting spectrogram has a height of 12 pixels and a width of (track length beats / granule_size_beats)

[0068] In a final step S73 an image processing feature detection network is cross-trained to identify sections in harmonic data. The spectrogram representation converts the dynamics and structure of a track into an image, that can be explored by an AI tool or pattern recognition algorithms.

[0069] The training of the convolutional neural network used in the procedure of Fig. 7, is performed by providing training data sets to the neural network along with human verified labels and timestamps identifying the parts and the granule ranges for each part, as shown in Table 6. Suitable music data sets are commercially available, for example SALAMI. Instead, or in addition, proprietary labelled data sets may be used.

[0070] Fig. 8 illustrates the function of Tabu Search as an example of an unsupervised optimization technique that may be used for mapping in the context of the present invention. As explained above, Tabu Search does not require training data, and will optimize the solution within the local space created by the parameters. In this case, the parameters include a base analysis 81 and an incoming analysis 82.

[0071] The system first generates, based on the base analysis 81 and the incoming analysis 82, an initial solution 83 that is a null transformation, meaning that no mutations have taken place.

[0072] Random mutations are then created by a mutation generator 84 and added to this solution to create a new solution 85. The fitness of the new solution, expressed as an error, is evaluated in a fitness evaluator 86, which will be described in more detail below. The output from step 86 is a scored solution 87, which is passed from the fitness evaluator 86 to a solution selector 88, which has stored one or more previous selected solutions. Based on the scores of this solution and the previous solutions, the solution selector 88 selects this new scored solution 87 or one of the previously stored solutions to be processed further. In both cases, the selected scored solution, either the new scored solution or one of the previously stored solutions is returned to the mutation generator 84 for further mutations.

[0073] The mutation generator 84 keeps a list of recently generated solutions of a fixed length (parameter max_sol_list length), and will not recreate a solution that exists in the list of recently generated solutions. This has the effect of causing the mutation generator to keep mutating until it finds new solutions, helping to avoid getting stuck in local minima.

[0074] The solution selector 88 keeps track of the best scoring solution, as well as a record of how many solutions have been evaluated since the last best scoring solution was found. When a certain number of solutions (parameter max_sol_since_best) have been generated without finding a new best solution, the last 'best solution' is selected as

the final solution 90.

[0075] Parameters `max_sol_since_best` and `max_sol_list_length` are determined via experimentation.

[0076] The function of the fitness evaluator 86 is illustrated in Fig. 9, using intensity analysis as an example. Input data include a first intensity evaluation 9a) of the original track and a second intensity evaluation 9b) of the incoming stem. Each bar `g`, `g'` in the diagram represents one granule of the respective track or stem. For simplicity, only some of the granules are labelled. The evaluation of the alignment is done by applying the computed mutations from the mutation generator shown in c) as non-straight arrows to the incoming stem analysis b) resulting in an mapping d) of the mutated incoming stem. In this example, the first four granules of the incoming stem are treated as a first segment and the last eight granules are treated as a second segment. The mapping changes the order of the granules, which in this case consists of moving the first segment of the incoming stem to the end of the stem.

[0077] Next, the error of the mapping against the base track stem analysis is calculated. The error e) represents the difference in intensity between the mapping d) and the base track a), and is shown in the lowermost line of Fig. 9 as the Euclidean distance between each granule of the base track stem analysis and the estimated mutated incoming stem analysis. The overall fitness is calculated as the sum of the Euclidean distances between each value, added to the number of mutations that were made to the incoming track which is multiplied by a weighting factor that is determined through experimentation, but which may be exposed as a user-controllable parameter allowing the user to influence the dynamics of the mapping algorithm. In other words, the aim of the method is to create a mapping of the incoming track that results in the smallest error, or difference between the mapping and the base track, or at least to keep the error at an acceptable level. It might be preferred to achieve this by the smallest possible number of mutations. In the example shown in Fig. 9, the mapping only has one mutation, swapping the first four granules of the incoming stem to the end of the stem.

[0078] Self-similarity matrices describe a data set by breaking it up into discrete granules, with a row and a column for each granule, and then analyzing the level of similarity between each granule and each other granule.

[0079] Such analysis is especially useful in the context of musical analysis when applied to harmonic information, as musical structure tends to be characterised by changes in harmonic patterns that occur throughout a piece of music. Therefore, the use of self-similarity matrices enables the comparison of the harmonic structures of two pieces of music. Considering the following simple example of one-note musical analysis, the string of notes, 'A A B A A B C C C B C C C B' described in a self-similarity matrix is shown in Table 7.

Table 7, a self-similarity matrix example

	A	A	A	B	A	A	A	B	C	C	C	B	C	C	C	B
A	1	1	1	0	1	1	1	0	0	0	0	0	0	0	0	0
A	1	1	1	0	1	1	1	0	0	0	0	0	0	0	0	0
A	1	1	1	0	1	1	1	0	0	0	0	0	0	0	0	0
B	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1
A	1	1	1	0	1	1	1	0	0	0	0	0	0	0	0	0
A	1	1	1	0	1	1	1	0	0	0	0	0	0	0	0	0
A	1	1	1	0	1	1	1	0	0	0	0	0	0	0	0	0
B	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1
C	0	0	0	0	0	0	0	0	1	1	1	0	1	1	1	0
C	0	0	0	0	0	0	0	0	1	1	1	0	1	1	1	0
C	0	0	0	0	0	0	0	0	1	1	1	0	1	1	1	0
B	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1
C	0	0	0	0	0	0	0	0	1	1	1	0	1	1	1	0
C	0	0	0	0	0	0	0	0	1	1	1	0	1	1	1	0
C	0	0	0	0	0	0	0	0	1	1	1	0	1	1	1	0
B	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1

[0080] In table 7, the track flows from the top left corner to the bottom left across the diagonal. In this description, the two distinct 'parts' of the musical string become distinctly visible in the self-similarity data, in a way that is abstracted from the actual musical data (e.g. specific notes).

[0081] As will be understood from Fig. 1, there may be several users 19, having systems comprising units 16, 17, 18, 20, and optionally 21, and all connected to the same analysis system comprising units 11 - 15. These users will all have access to the same tracks and stems, and also analysis data for the tracks and stems and will be able to share mappings with each other without sharing the actual music tracks.

Claims

1. An arrangement for creating a music mix based on at least a part of a base music track and one or more incoming music tracks, the arrangement comprising a stem remap unit (16), adapted to:
 - obtain a base analysis of the base music track, the base analysis comprising a number of descriptors with respect to its characteristics over time,
 - obtain an incoming track analysis of at least one stem of each incoming music track, the incoming track analysis comprising a number of descriptors with respect to the characteristics of at least a first and a second plurality of granules of one stem of the incoming track over time,
 - based on a comparison of the descriptors of the base music track and the at least one stem of each incoming music track, rearrange the order of a plurality of granules of the at least one stem to match the characteristics of the base music tracks over time, the rearranging resulting in a first mapping (20) of each stem to the base music track, the arrangement further comprising
 - a player unit (17) configured to obtain at least the incoming music track from an original track database (13), render a first output track to be played based on the at least one stem, and the at least one mapping.
2. An arrangement according to claim 1, wherein the player unit (17) is further configured to obtain the base music track, and render the first output track based on the at least one stem of an incoming track, the at least one mapping and at least one stem of the base music track.
3. An arrangement according to claim 1 or 2, wherein the granules are rearranged so that a number of consecutive granules form a segment and the mapping involves rearranging the order of at least a first and a second segment.
4. An arrangement according to any one of the preceding claims, further comprising a mutation generator (84) arranged to generate at least a first mutation of the first mapping to create a new mapping of the stem to the base music track, a selection unit (86) arranged to select the first mapping or the new mapping.
5. An arrangement according to any one of the claims 1-4, further comprising a suggestion engine (21) arranged to obtain a base analysis of the base music track, the base analysis comprising a number of descriptors with respect to its characteristics over time, select an alternative incoming music track and obtain an alternative incoming track analysis of at least one stem of the alternative incoming music track, the incoming track analysis comprising a number of descriptors with respect to the characteristics of at least a first and a second plurality of granules of one stem of the alternative incoming music track over time, and generate a suggested mapping based on the base music track analysis and the alternative incoming track analysis.
6. An arrangement according to claim 5, wherein the player unit (17) is arranged to render the output track as a combination of the first mapping and the suggested mapping.
7. An arrangement for providing music tracks suitable for mixing, comprising
 - a mastering and stem splitting unit (12) arranged to receive a first music track and split it into at least a first and a second stem, creating an original track,
 - an analysis unit (14) arranged to derive descriptors from the original track and/or stems, the descriptors relating to how the qualities of the original track and/or stem change over time,
 - an analysis database (15) arranged to store the result of the analysis,
 - an original track database (13) arranged to store the first music track in the form of stems, in a format that may be used to create a playable track.
8. A computer-based method of mixing at least parts of a base music track and one or more incoming music tracks, comprising

- obtaining, from an analysis database (15) an analysis of the base music track comprising a number of descriptors with respect to its characteristics over time
- obtaining, from the analysis database (15) an analysis of at least one stem of each incoming music track, comprising a number of descriptors with respect to its characteristics over time and a division of the at least one stem of the incoming music track into a number of granules,
- based on a comparison of the analyses of the base music track and the at least one stem of incoming music track, rearranging the granules of the at least one stem to match the characteristics of the base music track over time, the rearranging resulting in a first mapping of at least some granules of the stem,
- obtaining at least the base music track and each incoming music track, the incoming music tracks being split into at least two stems representing different components of the respective music track, and
- rendering an output track to be played, based on, the at least one stem of each incoming music track, and the first mapping of the at least one segment of each incoming track.

9. A method according to claim 8, wherein the output track is rendered based on the at least one stem of each incoming music track, and the first mapping of the at least one segment of the incoming track and at least one stem of the base music track.

10. A method according to claim 8 or 9, further comprising creating at least a first and a second segment, each segment comprising a plurality of consecutive granules, wherein the step of rearranging the granules involves rearranging the at least first and second segment.

11. A method according to any one of the claims 8 - 10, further comprising the step of repeating the method of claim 8 comprising based on a comparison of the analyses of the base music track and the at least one stem of the incoming music track, rearranging the granules of the at least one stem to match the characteristics of the base music track over time, the rearranging resulting in a second mapping of at least some granules of the stem which is different from the first mapping, obtaining at least the base music track and the incoming music track, the incoming music track being split into at least two stems representing different components of the respective music track rendering an output track to be played, based on at least one stem of the base music track, the at least one incoming track, and the second mapping of the at least one segment of the incoming track.

12. A method according to claim 11, wherein the step of rendering the output track to be played involves applying the first mapping and the second mapping at different times during playback of the output track.

13. A computer program product comprising computer-readable code means which, when executed in a computer will cause the computer to perform the method according to any one of the claims 8 - 12.

14. A player for use in an arrangement according to any one of the claims 1 - 7, said player being arranged to obtain a mapping of at least a first incoming stem, said mapping representing an order of segments of the at least first incoming stem, obtain the first incoming stem, render a reordered version of the stem based on the mapping divide the stem into chunks to facilitate playback of the stem according to the mapping, optionally together with one or more stems from the base track.

15. A player according to claim 14, configured to instantly switch from a first to a second version of audio based on the same base track, by determining a current beat position in the base track, and switching from a first beat position in the first version to a second beat position in the second version, both the first and the second beat position corresponding to the current beat position in the base track, to maintain musical continuity.

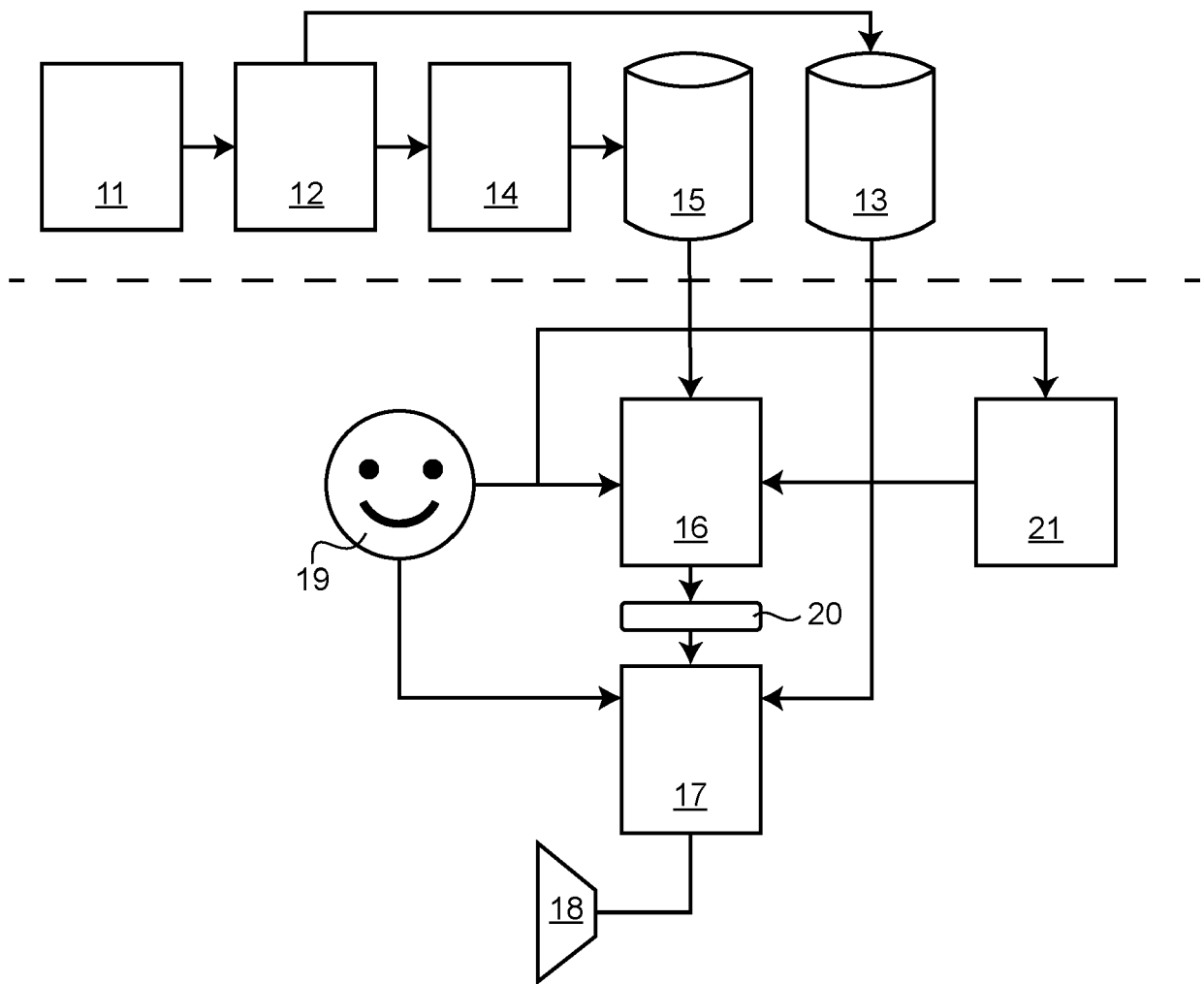


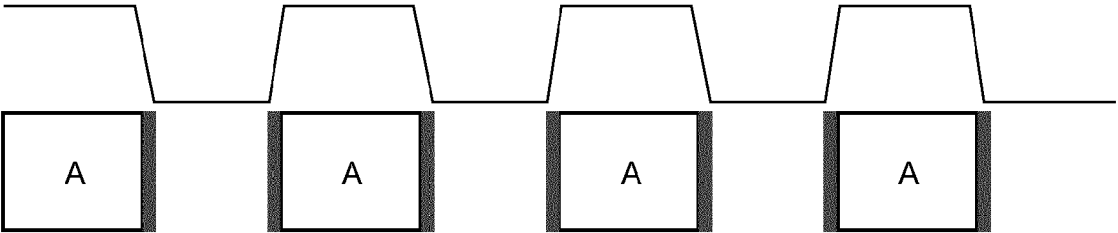
Fig. 1



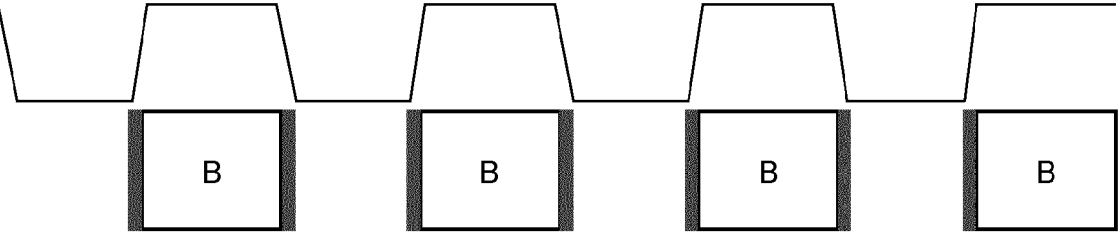
a)



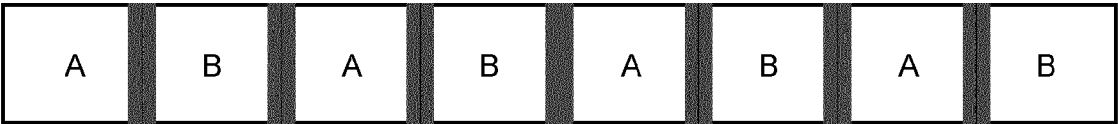
b)



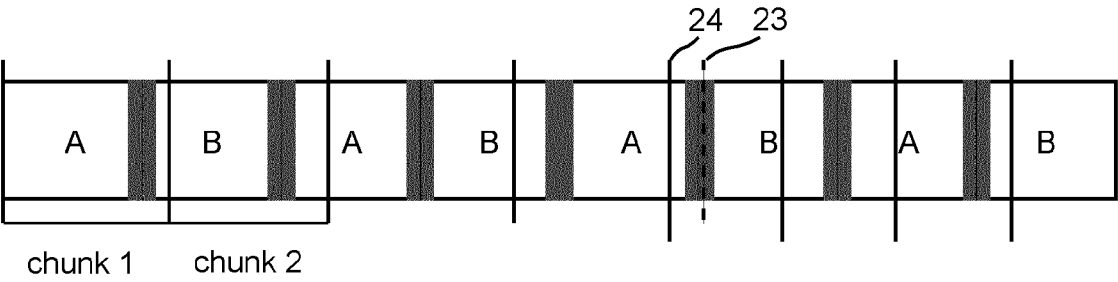
c)



d)



e)



f)

Fig. 2

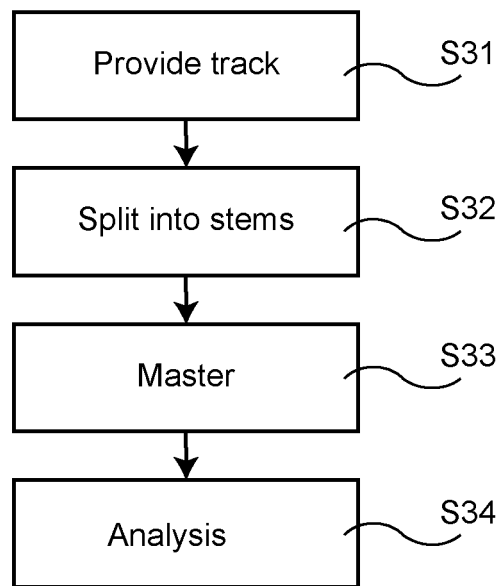


Fig. 3

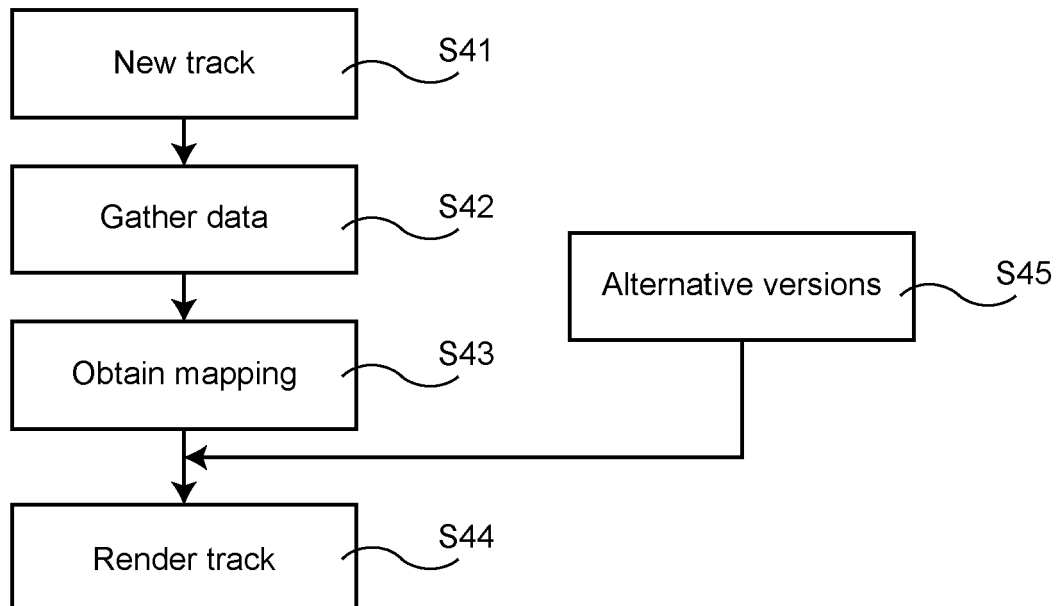


Fig. 4

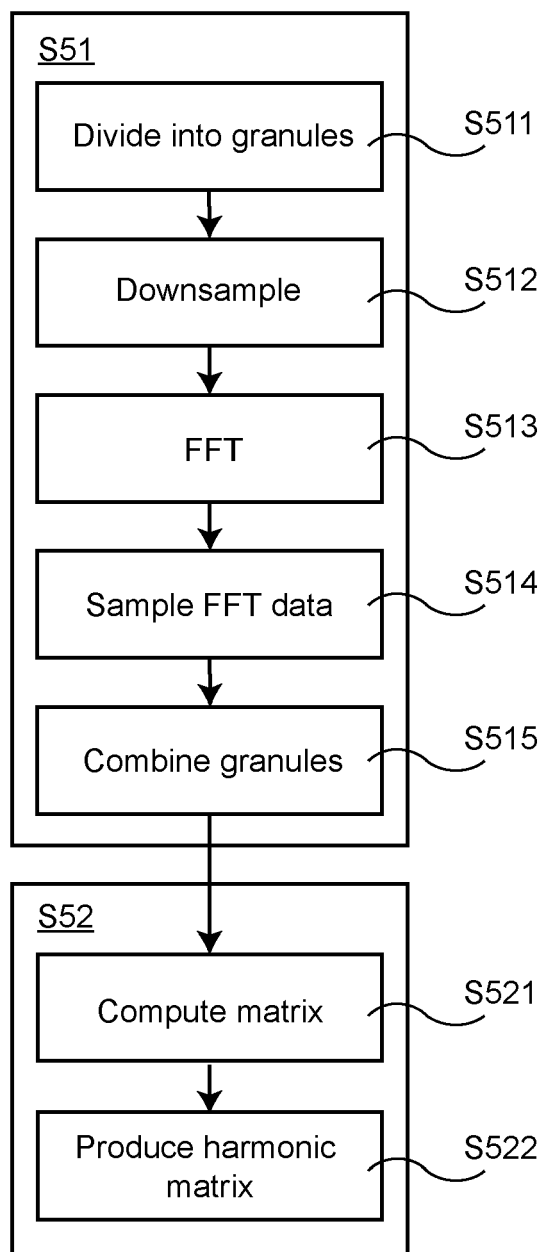


Fig. 5

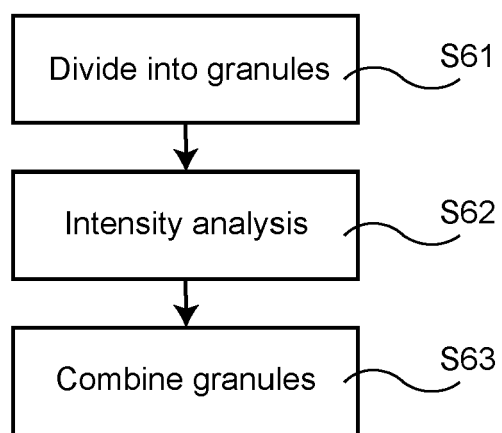


Fig. 6

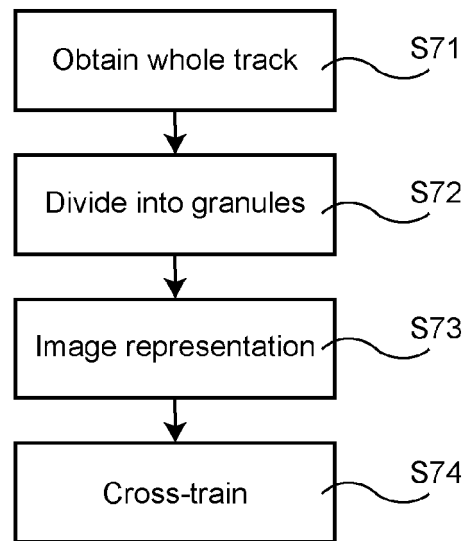


Fig. 7

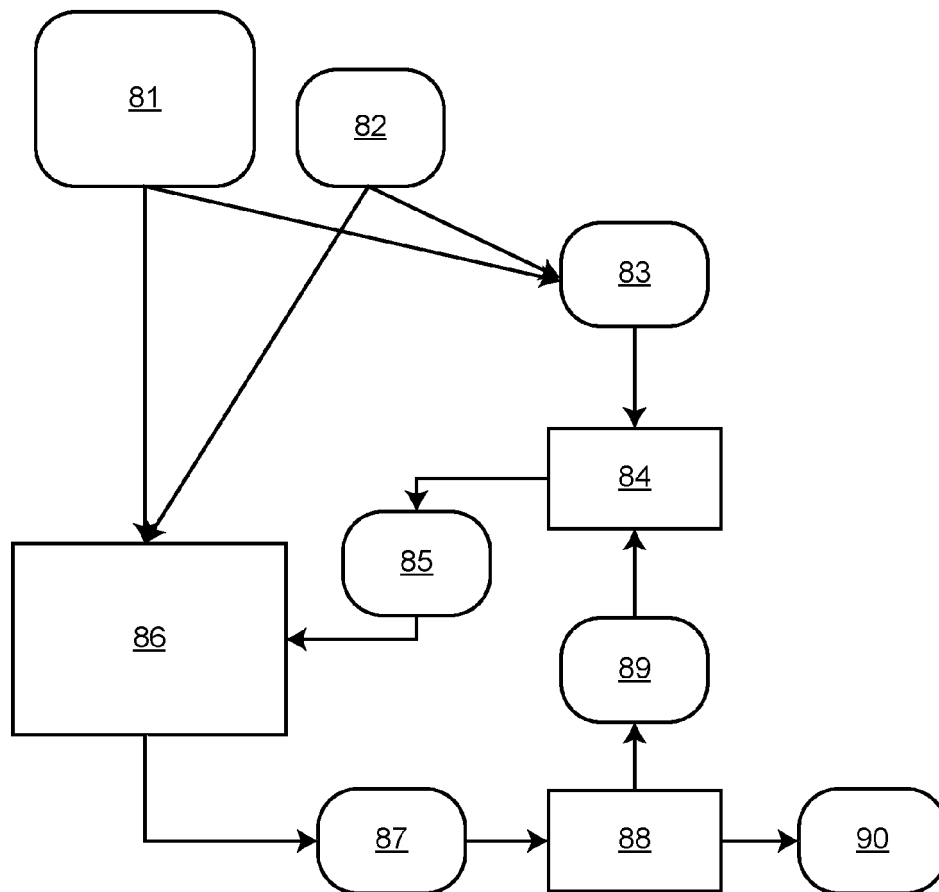


Fig. 8

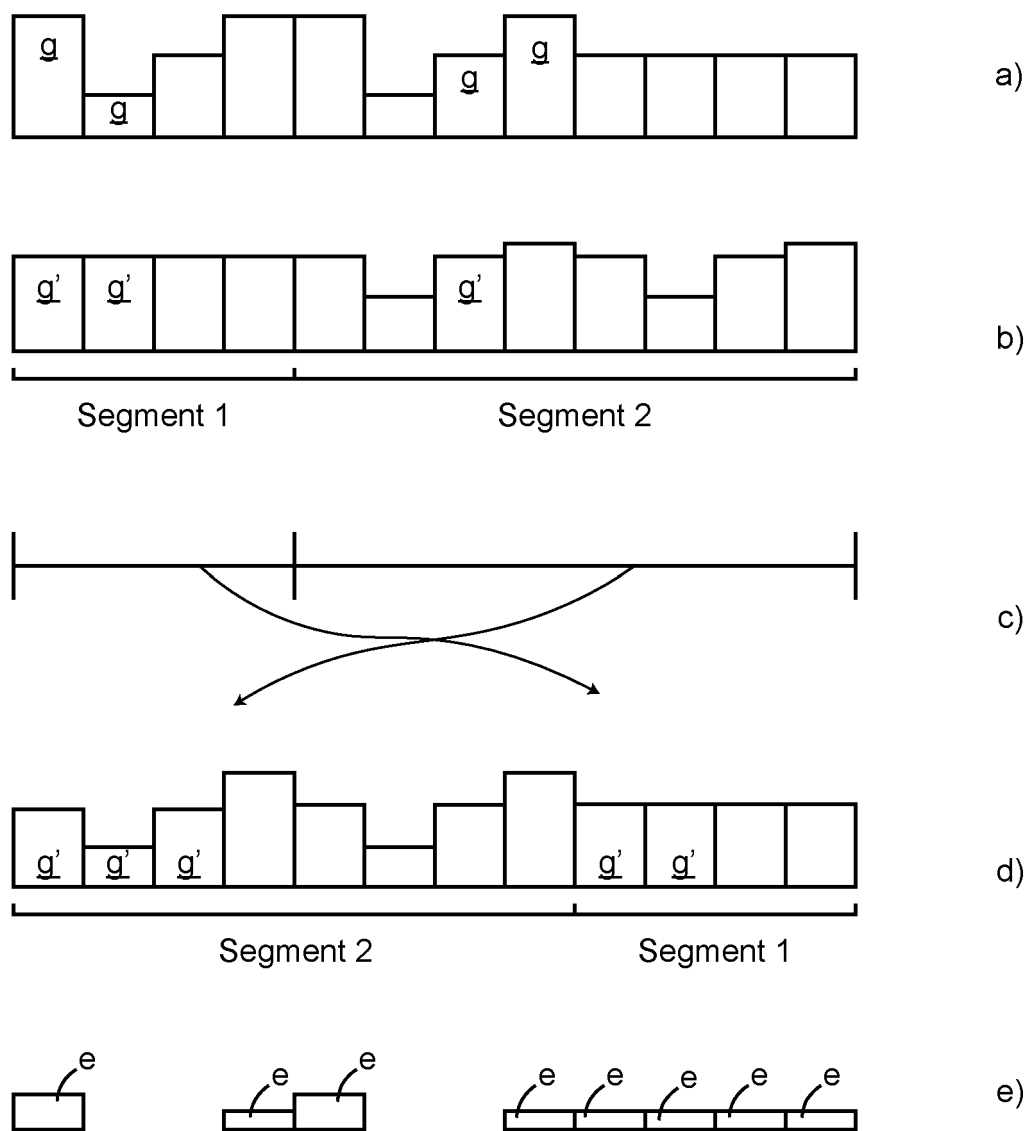


Fig. 9



EUROPEAN SEARCH REPORT

Application Number

EP 24 18 8536

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	US 2013/170670 A1 (CASEY MICHAEL [US]) 4 July 2013 (2013-07-04)	1,2,4, 7-9,13, 14	INV. G10H1/00
Y	* paragraph [0004] - paragraph [0022] * * paragraph [0023] - paragraph [0028]; claims 1,2; figure 1 * * paragraph [0053] - paragraph [0058]; figure 4 * * paragraph [0059] - paragraph [0073]; figure 5 *	3,5,6, 10-12,15	
Y	----- US 2023/075074 A1 (BOSCH VICENTE JUAN JOSÉ [FR] ET AL) 9 March 2023 (2023-03-09) * paragraphs [0003], [0074], [0143] - paragraph [0144]; figure 18 * * paragraph [0115] - paragraph [0117]; figure 4 * * paragraphs [0084], [0137], [0167]; claims 29,42,37; figures 1,20, 26 * * paragraph [0118] - paragraphs [0121], [0145] - [0157]; claim 36; figure 8 *	5,6,11, 15	
Y	----- US 2020/357371 A1 (REIN DIETER [DE] ET AL) 12 November 2020 (2020-11-12) * paragraph [0027] *	12	TECHNICAL FIELDS SEARCHED (IPC) G10H H04N
Y	----- LU L ET AL: "Audio Textures: Theory and Applications", 1 March 2004 (2004-03-01), IEEE TRANSACTIONS ON SPEECH AND AUDIO PROCESSING, IEEE SERVICE CENTER, NEW YORK, NY, US, PAGE(S) 156 - 167, XP011110595, ISSN: 1063-6676 * page 139, right-hand column, paragraph IV-A; figures 5, 5(d) *	3,10	
The present search report has been drawn up for all claims			
Place of search Munich		Date of completion of the search 30 January 2025	Examiner Glasser, Jean-Marc
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 (P34C01)

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

EP 24 18 8536

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.

30-01-2025

10

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 2013170670 A1	04-07-2013	US 2013170670 A1	04-07-2013
		WO 2011103498 A2	25-08-2011

US 2023075074 A1	09-03-2023	EP 3843083 A1	30-06-2021
		US 2021201863 A1	01-07-2021
		US 2023075074 A1	09-03-2023

US 2020357371 A1	12-11-2020	EP 3907629 A1	10-11-2021
		EP 3937163 A1	12-01-2022
		US 2020357370 A1	12-11-2020
		US 2020357371 A1	12-11-2020

25

30

35

40

45

50

55

EPO FORM P0459

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

REFERENCES CITED IN THE DESCRIPTION

This list of references cited by the applicant is for the reader's convenience only. It does not form part of the European patent document. Even though great care has been taken in compiling the references, errors or omissions cannot be excluded and the EPO disclaims all liability in this regard.

Patent documents cited in the description

- US 20230360618 A1 [0004]
- US 20230360618 A [0004]