(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
23 January 2003 (23.01.2003)

PCT

(10) International Publication Number
WO 03/007112 A2

(51) International Patent Classification⁷: G06F

(21) International Application Number: PCT/US02/21770

(22) International Filing Date: 9 July 2002 (09.07.2002)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
60/303,846 9 July 2001 (09.07.2001) US

(71) Applicant *(for all designated States except US)*: **VISIBLE WORLD, INC.** [US/US]; 6th Floor, 527 West 34th Street, New York, NY 10001 (US).

(72) Inventors; and
(75) Inventors/Applicants *(for US only)*: **HABERMAN, Seth** [US/US]; Apt. 7C, 1 West 85th Street, New York, NY 10025 (US). **NIEMEIJER, Gerrit** [US/US]; 23 Arcularius Terrace, Maplewood, NJ 07040 (US). **BOLTZE, Thomas** [US/US]; Apt. 2706, 30 River Court, Jersey City, NJ 07310 (US). **JANSEN, Alex** [US/US]; Apt. 5J, 101 West End Street, New York, NY 10023 (US).

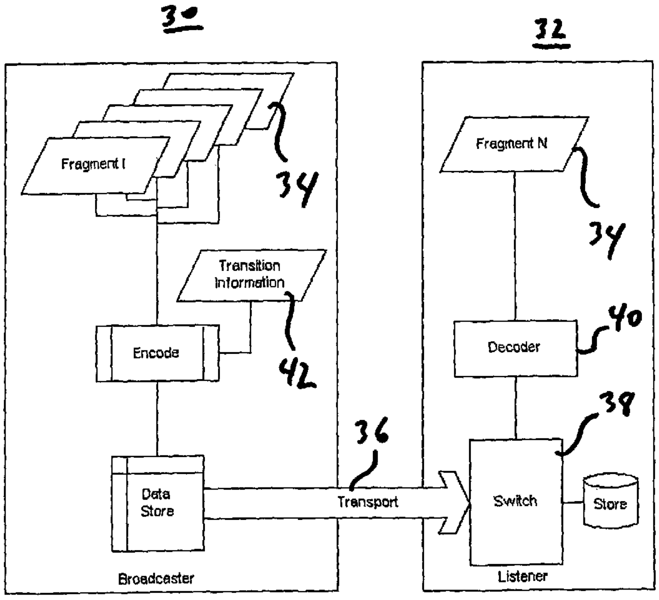
(74) Agents: **LOWRY, David, D.** et al.; Brown Rudnick Berlack Israels LLP, One Financial Center, Boston, MA 02110 (US).

(81) Designated States *(national)*: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, TZ, UA, UG, US, UZ, VN, YU, ZA, ZW.

(84) Designated States *(regional)*: ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE, SK, TR), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

[Continued on next page]

(54) Title: SYSTEM AND METHOD FOR SEAMLESS SWITCHING OF COMPRESSED AUDIO STREAMS



(57) Abstract: A system and method for seamless switching and concatenation of compressed audio streams in Internet, Digital Radio, Digital Television, DVD, storage, and other applications. The technology allows switching between streams at pre-determined points without the introduction of audible artifacts. It can be used for the personalization messages such as advertisements, news systems and other.

WO 03/007112 A2

Published:

— *without international search report and to be republished upon receipt of that report*

SYSTEM AND METHOD FOR SEAMLESS SWITCHING OF COMPRESSED AUDIO STREAMS

RELATED APPLICATION

This patent application claims priority to U.S. Provisional Application Serial No. 60/303,846 filed July 9, 2001 which is hereby incorporated by reference.

FIELD OF THE INVENTION

This invention is directed towards digital audio, and more particularly towards a method for preparation and compression of individual audio fragments that allows for seamless playback of sequences of such fragments.

BACKGROUND

Co-owned U.S. patent application 09/545,015 (which is incorporated herein by reference) describes a system and method for creating personalized messages (such as personalized advertisements and personalizes news). An example of a personalized message structure 20 is shown in Fig. 1. It starts with a common opening 22, followed by three possible options for the middle part 24 and a common closing 26. One instance of this message is given by the sequence opening then option 1 then closing; another instance is given by opening then option 2 then closing.

A personalized audio message structure as depicted in Fig. 1, is typically created by an audio designer using dedicated tools. The audio fragments in the message structure are typically generated by the audio designer using editing tools such as, but not limited to, AVID MediaComposer, ProTools, etc.

Having the personalized message structure as well as the associated audio fragments available, a switching device can create an instance of the personalized message by playing the proper fragments in sequence.

The personalized message structure and the associated audio fragments can be made available to the switching device in a variety of ways.

In one specific scenario, the audio fragments part of the personalized message will be broadcast in compressed form in different digital television channels and assembled by a switching device, such as a digital set-top-box, at the listeners location to form one specific instance of the message. One way in which the instance can be assembled is by switching channels on-the-fly at the moment a transition from one fragment to another must be made.

In another specific scenario, the media fragments will be made available to a switching device with storage (e.g., a DVD player, a PC) using a storage medium, such as a CD-ROM or a DVD disk. The fragments will be stored on this storage medium in compressed form. The switching device will select and load the proper fragments from the storage medium, and play them in sequence.

However, current compression technology applied in digital radio, digital TV, Internet and storage applications, including MPEG and AC-3 encoding and compression, does not readily allow for seamless concatenation or switching of compressed audio fragments, which poses a major problem.

One reason for this problem is that most audio codecs used in the domains of digital television, DVD, Internet streaming, and others operate on frames (fixed size groups) of samples, instead of individual samples. One frame, which is a number of consecutive audio samples, is encoded and decoded as a unit and cannot be broken into smaller subunits. Consequently, once the material is encoded, a transition or switch between options can occur only on frame boundaries. As typically used in the digital television domain, a codec for MPEG Layer II has a frame length of 1152 samples. A codec for Dolby AC-3 has a frame length of 1536 samples. If the length of a fragment (in samples) to be compressed is not an exact multiple of the frame size (in samples), the remainder of the fragment will either be thrown away during encoding, leading to loss of data and severe glitches, or it will be padded with zeroes, leading to pauses in the presentation. Obviously both are disadvantageous as they lead to a non-seamless presentation when concatenating and playing audio options after decoding.

Another reason for the problem is that most audio codecs used in the domains of digital television, DVD, Internet streaming, and others, encode audio frames based on the contents of previous frames.

In a filter-bank based codec, such as MPEG layer II, the outcome of the encoding process of a current audio frame depends on the filter bank states produced by the past frames. The filter bank acts like a memory. More specifically, MPEG Layer II uses a 32-band filter bank to decompose the incoming signal into sub band samples, which are then quantized. Alias cancellation affects neighboring sub bands, but not successive frames, so it does not pose a problem for the switching. However the states of the filter bank in the encoder and in the decoder depend on the previously encoded frame. To achieve perfect reconstruction after the decoder filter bank, the filter states must be the same as in the encoding process.

In a transform-based codec, such as AC-3, the window and overlap-add mechanism introduces a dependency between successive frames. Here the overlap-add requires consecutive frames to be encoded and decoded in the right context to ensure that alias components cancel out in time. More specifically, AC-3 uses a windowing of the input data, a DCT and subsequent IDCT and overlap-add in the decoder. Successive windows overlap. Alias cancellation is in the time domain and requires the proper history to work. If arbitrary AC-3 streams are concatenated, the alias cancellation does not work at the splice point. This leads to audible artifacts, which are theoretically much worse than in the MPEG case. At the start of an encode process of several frames a start window is used which effectively mutes the first 256 samples of the first frame. This creates a clearly audible gap, which is not acceptable for concatenation. The last frame of a decoded sequence ends with a fade out of the signal over the final 256 samples; due to the missing overlap add of the next frame.

The fact that most audio codecs use a history means that fragments that are intended to be played back in sequence cannot be encoded in isolation, even if their lengths are exact multiples of the frame size defined by the compression scheme. If no additional measures are taken, the transition from one fragment to another will not be seamless, and lead to audible artifacts.

Accordingly, what is required is a method and system for manipulating and encoding/compressing audio fragments such that a switching device can decode and play such compressed fragments in sequence without audible gaps or artifacts. The present invention discloses such a method and system.

SUMMARY

The technology described in the present application addresses the issues around seamless playback of sequences of separately encoded and compressed, digital audio fragments.

The present invention provides for a method and system for manipulating audio fragments and subsequently encoding/compressing such audio fragments in a manner that allows for seamless playback at a switching device, thus providing a seamless, uninterrupted, presentation to the listener.

The manipulation of the audio fragments according to the present invention comprises aligning beginning and end times of audio options in a personalized message on frame boundaries, where the frame size is defined by the compression scheme to be used (e.g., 1152 for MPEG Layer II and 1536 for AC-3).

The encoding of the audio fragments according to the present invention takes history into account for example by prepending one additional audio frame at the start of an fragment to set the history of the encoder. This frame is subsequently discarded from the compressed result since it is only used to initialize the history of the encoder. The audio frame to be prepended is obtained from the end of one of the options that can directly precede the fragment to be encoded.

An illustrative embodiment of the present invention is used to process and encode the audio fragment, also called options, in a personalized message (which can be an advertisement, a news program,...). This allows a receiver, such a digital set-top box, to seamlessly, and on-the-fly, assemble and play out one instance of the message while the various message options are provided to the set-top-box using an MPEG-2 transport stream.

An advantage of the present invention is the ability to manipulate and encode audio fragments belonging to a personalized message structure such that playout of instances of the message will be seamless, i.e., without audible artifacts, at all points of the message, including around the transition points between audio fragments.

Another advantage of the present invention includes the preparation of an personalized message for efficient transport and distribution over digital television channels, DVDs, and other distribution means.

An illustrative embodiment of the present invention includes a method of preparing a plurality of digital audio fragments to allow switching between at least one source fragment and at least one target fragment. The method includes aligning an end of at least one source fragment with a beginning of at least one target fragment, for all possible valid combinations of at least one source fragment and at least one target fragment; wherein the at least one source fragment is aligned to be a length that is an exact multiple of a predetermined number. The method also includes moving a sequence of audio samples from a digital audio fragment which was shortened because of the alignment step, to a plurality of digital audio fragments which were lengthened because of the alignment step. The moved sequence of audio samples is a length which will result in at least one source fragment to be a length that is an exact multiple of the predetermined number. Typically, the predetermined number is a frame size.

The illustrative embodiment also includes moving a sequence of audio samples from the end of at least one source fragment to the beginning of at least one target fragment; wherein the sequence of audio samples is a length which will shorten the one source fragment to be a length that is an exact multiple of the predetermined number.

The present invention also includes copying a last frame of a source fragment to the beginning of at least one target fragment, compressing the at least one target fragment using a compression scheme which uses frames and wherein subsequent frame encoding depends upon an encoding of at least one previous frame. The method includes removing data from the beginning of the compressed at least one target fragment, the data corresponding to a first frame of the at least one target fragment.

An embodiment of the present invention includes a system for preparing a plurality of digital audio fragments for transmission to allow a switching device to switch between at least one source fragment and at least one target fragment. The system includes an audio aligner module, coupled to a source of the plurality of audio fragments, to align beginning and ends of the plurality of audio fragments to selected times based on an exact multiple of a predetermined number; and an audio compression module, coupled to the audio aligner module, to compress the plurality of audio fragments as a sequence of frames, wherein each frame comprises a sequence of audio samples; and the length of the frame is the predetermined number. The system works for audio fragments that are transmitted using any one of several transport mechanisms, including MPEG compliant, digital television, dvd broadcast, dvd storage, CD ROM, and internet.

BRIEF DESCRIPTION OF THE DRAWINGS

The foregoing and other features and advantages of the present invention will be more fully understood from the following detailed description of illustrative embodiments, taken in conjunction with the accompanying drawings in which:

Fig. 1 illustrates an example of a (simple) personalized message structure;

Fig. 2 illustrates a processing model for personalized messages;

Fig. 3A-3C illustrate the three possible situations for transitions between options in a personalized message (branch, confluence, and junction);

Fig. 3D illustrates an example of a more complex personalized message structure, containing various types of transitions;

Fig. 4 illustrates an example branch situation with the transition point not aligned on a frame boundary;

Fig. 5 illustrates the transition of Fig. 4 moved to an earlier frame boundary in accordance with the present invention;

Fig. 6 illustrates providing a temporary copy of the last frame of the source option for encoding target options in the transition of Fig. 5 in accordance with the present invention;

Fig. 7 illustrates an example confluence situation with the transition point not aligned on a frame boundary;

Fig. 8 illustrates the transition of Fig. 7 moved to a later frame boundary in accordance with the present invention;

Fig. 9 illustrates the transition of Fig. 8 moved one frame later to realize identical endings of all source options, in accordance with the present invention;

Fig. 10 illustrates providing a temporary copy of the last frame of one of the source options for encoding target options in the transition of Fig. 9 in accordance with the present invention;

Fig. 11 illustrates an example case of a junction where the transition is not located on a frame boundary, and where the ending of all source options is identical, and where the beginning of all target options is identical;

Fig. 12 illustrates the transitions of Fig. 11 moved to a later frame boundary in accordance with the present invention;

Fig. 13 illustrates providing a temporary copy of the last frame of any source option for encoding target options in the transition of Fig. 12 in accordance with the present invention;

Fig. 14 illustrates an example case of a junction where the transition is not located on a frame boundary, and where the ending of all source options is different, and where the beginning of all target options is identical;

Fig. 15 illustrates the transition of Fig. 14 moved to a later frame boundary in accordance with the present invention;

Fig. 16 illustrates providing a temporary copy of the last frame of any source option for encoding target options in the transition of Fig. 15 in accordance with the present invention;

Fig. 17 illustrates an example case of a junction where the transition is not located on a frame boundary, and where the ending of all source options is identical, and where the beginning of all target options is different;

Fig. 18 illustrates the transition of Fig. 17 moved to an earlier frame boundary in accordance with the present invention;

Fig. 19 illustrates providing a temporary copy of the last frame of any source option for encoding target options in the transition of Fig. 18 in accordance with the present invention;

Fig. 20 illustrates an example case of a junction where the transition is not located on a frame boundary, and where the ending of all source options is different, and where the beginning of all target options is different;

Fig. 21 illustrates the transition of Fig. 20 moved to a later frame boundary in accordance with the present invention; and

Fig. 22 illustrates the transition of Fig. 20 moved to an earlier frame boundary in accordance with the present invention.

DETAILED DESCRIPTION

A processing model capable of supporting seamless switching/concatenation of compressed audio fragments in accordance with the present invention is shown in Fig. 2. At the source 30 the audio fragments 34 are encoded and prepared for transport. At the listener location 32 the fragments are received, and a subset of them is decoded and played out in sequence. The fragments 34 are individually encoded with the structure of the possible transitions in the personalized message 42 taken into account, and are optionally stored. The compressed fragments 34 are transported via a channels 36 to the listener 32. This transport may occur in real-time (as in TV broadcasts) or in non real-time (as in storage media such as

DVD). The fragments are optionally stored at the listener location 32 before decoding and playout.

The compressed files to be decoded and played are selected by a switch 38, and provided to the decoder 40 in sequence. The decoder 40 decompresses the resulting bit stream and presents the audio to the listener.

The different scenarios for transitions in a personalized message are Branches Fig. 3A, Confluences Fig. 3B, and Junctions Fig. 3C. Branches consists of one option (source) transitioning to a plurality of options (targets). Confluence consists of multiple options (sources) transitioning to one subsequent option (target). A junction consists of multiple options (sources) transitioning to subsequent multiple options (targets). It is important to note that the options at the two sides of the branch do not need to have identical lengths. The only requirement on the transition is that the source options end at the exact same time, and that the target options begin at the exact same time. This is further illustrated in Fig. 3D, which shows an example of a personalized message structure with 10 options and 5 transitions. Arrows in the Figure denote which options can be played in sequence as identified by the creator of the personalized message. It is interesting to note that there is a 1 to 1 transition in the example. This can either be treated as a branch or a junction. Furthermore it is interesting to note that the transitions are defined based on the information from the creator which options can be played in sequence. This leads, for instance to two different transitions between options 0,2,3 and 4,5,6, i.e., a branch from 3 to 5,6 and a confluence from 0,2 to 4.

The individual steps that need to be taken from the original (uncompressed) options and message structure to the final encoded (compressed) options and message structure will now be provided. Several of the disclosed steps might be combined into one physical step, or certain steps might be split-up into smaller physical steps, but the focus here is on the conceptual steps rather than those implemented as separate entities. Also, no assumptions are being made on where a conceptual step is executed by a human or a machine, since both are possible.

The first step in personalized messaging involves the creation of the uncompressed audio options and their possible transitions. The creation process as assumed in the present invention provides full flexibility to the creator of the personalized message with respect to the exact time of and type of possible transitions between options. It is understood that the creator could already perform some of the steps as described below in a manual fashion while creating the uncompressed audio options, but this is not required.

In order to avoid clicks and pops when playing sequences of uncompressed audio according to the personalized message structure, the audio on both sides of each transition must form smooth continuous waveforms. All allowed transitions must sound smooth in the uncompressed domain, otherwise the switch in the compressed domain cannot be done without at least the same audible artifacts. Hence, playback of the compressed options will only be seamless when playback of the uncompressed material is seamless.

The first constraint on a personalized message whose options have to be compressed is related to audio frames (also called 'Access Units' in MPEG). Most audio codecs used in the domains of digital television, DVD, Internet streaming, and others operate on frames of samples at any one time. One frame, which is a number of consecutive audio samples, is encoded and decoded as a unit and cannot be broken into smaller subunits. Consequently, once the material is encoded, a transition or switch between options can only occur only on frame boundaries. Frame lengths for codecs are usually defined as number of samples rather than duration, leading to different durations for different sample rates. As typically used in digital television, a codec for MPEG Layer II has a frame length of 1152 samples. A codec for Dolby AC-3 has a frame length of 1536 samples.

Thus, transitions in the personalized message structure as defined during creation need to be adjusted such that they occur on audio frame boundaries, rather than on arbitrary audio samples. The adjustment of transitions to frame boundaries (meaning that each option in the message has a length that is an exact multiple of the frame size of the coding scheme to be used) is required before the options themselves are actually encoded. This is needed to avoid playout artifacts, since encoders operate on a frame-by-frame basis. Parts of frames cannot be processed, and will either be thrown away, leading to loss of data and severe glitches, or they will be padded with zeroes, leading to pauses in the presentation. Obviously both are unwanted as they lead to a non-seamless presentation when concatenating compressed audio options.

The present invention ensures that transitions are correctly moved to audio frame boundaries with as little as possible loss of information (audio data). For each of the three types of transitions a different scheme is disclosed that moves the transition to the closest earlier or later frame boundary.

The next constraint related to compression of options in personalized messages is history. Most audio codecs used in the domains of digital television, DVD, Internet streaming, and others, encode audio frames based on the contents of previous frames. In a filter-bank based codec, such as MPEG layer II, the outcome of the encoding process of a current audio frame depends on the filter bank states produced by the past frames. The filter bank acts like a memory. In a transform-based codec, such as AC-3, the window and overlap-add mechanism introduces a dependency between successive frames. Here the overlap-add requires consecutive frames to be encoded and decoded in the right context to ensure that alias components cancel out in time.

The present invention ensures that the history of the encoders and decoders is maintained correctly across transitions. Common to all transitions is that typically one additional audio frame from a preceding option is encoded at the start of an option to set the history of the encoder. This frame is discarded after the encoding is done, resulting in the compressed version of the option. This will be disclosed further below.

Thus, processing the options in a personalized message after creation in accordance with the present invention can be split in two consecutive steps: Alignment and Encoding. Each of these steps is disclosed below.

Alignment

The first step, alignment, will ensure that all options in the personalized message have a length that is an exact multiple of the frame length of the intended compression scheme. This allows encoding of options without the encoder having to either discard data or introduce silence. Thus, during alignment, transitions between options are moved to frame boundaries.

The second main function of alignment is ensuring that all source options in each transition have an as similar last frame as possible. The reason for this is that the last frame of one of these options will be used during encoding of the target options in the transition to initialize the encoder history buffer, as disclosed below in the section on encoding. Since only one source option can provide the frame to be used to fill the encoder buffer, transitions from source options to target options will only be perfectly seamless when the last frames of all source options in a transition are identical.

Alignment of the complete personalized message is done one single transition at a time. A transition can be handled the easiest when the source options of that transitions start at a frame boundary, otherwise the transition might have to be revisited/reprocessed later on in the process. This means that transitions are handled preferably in a time-increasing fashion, meaning that later transitions are handled after earlier transitions. This way, it is assured that, when handling one transition, all source options in that transition always start on a frame boundary.

Any person skilled in the art can see that different methods/orders of aligning a template are also possible. The one just described is one example that is particularly easy to implement in specific embodiments of this invention.

In the following section is described how one individual transition will be handled, assuming all transitions that happen before it in time in the message structure have already been handled as described below. We disclose the handling of each of the three different types of transitions separately:

- Branch (1 to M transitions)
- Confluence (N to 1 transitions)
- Junction (N to M transitions)

Branch

For branching, the transition point is moved to the closest earlier frame boundary. This is required because a move to the closest later frame boundary would lose audio samples from all target options but one. The audio samples from the source option that are between the new and the old transition point are appended to the beginning of each target option. As illustration, Fig. 4 shows the original situation for two target options. The original transition 52 as set by the creator of the personalized message is shown. It does not lie on a frame boundary 50. As a result, assuming that option n-1 starts on a frame boundary, the last samples 54 of option n-1 do not add up to a complete frame length.

In Fig. 5 the transition 52 is moved to the closest earlier frame boundary by removing the audio samples 54 of option n-1, and prepending them to the audio samples 56 of target options n and n+1. The exact same samples 54 are prepended to each of these two options. The transition 52 now occurs on an audio frame boundary 50, which makes it possible to switch seamlessly from option n-1 to option n or option n+1 after compression. Also, options n and n+1 now start on frame boundaries, allowing transitions in which they appear as source options to be treated as disclosed.

No further processing is needed for branching.

Confluence

For confluence, the transition point is moved to the closest later frame boundary. This is required because a move to the closest earlier frame boundary would lose audio samples from all source options but one. The audio samples from the target option that are between the old and the new transition point are appended to each source option.

As illustration, Fig. 7 shows the original situation for two source options. The original transition 52 as set by the creator of the personalized message is shown. It does not lie on a frame boundary 50. As a result of this, assuming that options n-1 and n start on frame boundaries (which can be different), the last samples of each of the options n-1 and n do not add up to a complete frame (they will be off by the same amount of samples since both start on a frame boundary).

In Fig. 8 the transition 52 is moved to the closest later frame boundary 50 by removing the audio samples 56 of option n+1, and appending them to the audio samples 54 of options n-1 and n . The same samples are appended to each of these two options. Transition 52 now occurs on an audio frame boundary, which makes it possible to switch seamlessly from option n-1 or option n to option n+1 after compression. Also, option n+1 now start on a frame boundary, allowing transitions in which it appears as source option to be treated as disclosed.

To allow encoding such that perfect seamless transitions between source and target options can be achieved an additional processing step is required for confluence. This step is to assure that the last frame of each source option is identical, required for optimal initialization of the history of the encoder buffer. Therefore, the first full audio frame in target option n+1 is removed from option n+1 and appended to options n-1 and n. This is illustrated in Fig. 9. The samples in the complete frame 60 as also shown in Fig. 8 have been moved from the target option to the end of each source option.

Junction

The case of a junction requires special attention, as previously discussed. For the required alignment of the transition on an audio frame boundary, this means that either ending audio samples from (any of) the source options must be removed and prepended to each of the target options, or that beginning audio samples from (any of) the target options must be removed and appended to each of the source options.

The decision whether to move audio from source to target options (which moves the transition to an earlier time) or from target to source options (which moves the transition to a later time) will depend on which leads to the least (or no) loss of audio data. We have two choices:

(a) Remove q samples from each source option and then prepend q samples to each target option. Here, q is the amount of samples needed to move the transition to the next earlier frame boundary.

(b) Remove r samples from each target option and then append r samples to each source option. Here, r is the amount of samples needed to move the transition to the next later frame boundary.

The following four scenarios exist.

Scenario 1: For each source option, its last q samples are identical to the last q samples of each other source option. Furthermore, for each target option, its first r samples are identical to the first r samples of each other target option.

In this case the transition is moved later in time, i.e., the first r samples from each target option are removed, and r samples are appended to each source option. No audio data is lost while moving samples between target and source options.

Moving the transition later in time is done because the last frame of samples (q+r is identical to the frame size) of each source option will now be identical, meaning that the last frame of each option that can be followed by a target option is identical, allowing for the perfect initialization of the history of the encoder.

Fig. 11 further illustrates this scenario. The last q samples 54 are identical for each of the source options n-2 and n-1. The first r samples 56 are identical for each of the target options n and n+1. The result, removing the first r samples from options n and n+1, and then appending one such segment of r samples (taken from either option n or n+1) to both options n-2 and n-1 is depicted in Fig. 12. As can be seen the transition 52 is now moved to the closest later frame boundary 50.

Scenario 2: For each source option, there is at least one other source option for which the last q samples are different between the two source options. Furthermore, for each target option its first r samples are identical to the first r samples of each other target option.

In this case the transition will also be moved later in time, i.e., the first r samples from each target option will be removed, and r samples will be appended to each source option. No audio data is lost while moving samples between target and source options. Note that the alternative, moving the transition earlier in time, would always lead to loss of audio data, which is disadvantageous.

Since r is always smaller than the frame size, the initialization of the history of the encoder will not be perfect, since the source options do not have a full frame of audio in common at the end. Therefore, for small values of r, audible artifacts might occur, depending on how much the last frames of the source options differ. If the last part of each source option is reasonably similar (which should be guaranteed by the creator of the personalized message), this will not lead to audible artifacts.

Fig. 14 further illustrates this scenario. The first r samples 56 are identical for each of the target options n+1 and n+2. The result, removing the first r samples from options n and n+1, and then appending one such segment of r samples (taken from either option n or n+1) to both options n-2 and n-1 is depicted in Fig. 15. As can be seen the transition 52 is now moved to the closest later frame boundary.

Scenario 3: For each source option its last q samples are identical to the last q samples of each other source option; Furthermore, for each target option, there is at least one other target option for which the first r samples are different between the two target options.

In this case the transition is moved earlier in time, i.e., the last q samples are removed from each source option, and q samples are prepended to each target option. No audio data is lost while moving samples between source and target options. Note that the alternative, moving the transition later in time would always lead to loss of audio data, which is disadvantageous.

After moving the samples, the source options will likely be different in their last frame. This means that the initialization of the audio history during encoding will be imperfect since the initialization can be done with the last frame from only one of the source options. If the last frames of all source options are reasonably similar (which should be guaranteed by the creator of the personalized message) this will usually not lead to audible artifacts.

Fig. 17 further illustrates this scenario. The last q samples 54 are identical for each of the source options n-1 and n-2. The result, removing the last q samples from options n-2 and n-1, and then appending one such segment of r samples (taken from either option n-2 or n-1) to both options n and n+1 is depicted in Fig. 18. As can be seen the transition 52 is now moved to the closest earlier frame boundary.

Scenario 4: For each source option, there is at least one other source option for which the last q samples are different between the two source options. For each target option, there is at least one other target option for which the first r samples are different between the two target options.

In this case some audio data will always be lost, no matter whether the transition point is moved to an earlier or later frame boundary. In specific embodiments of this invention, the following heuristics are used to decide in what direction to move the frame boundary:

(a) Move the transition in the direction that leads to removing the least number of audio samples, leading to the least amount of information that will be lost. This means, if q is smaller than r, the transition is moved to the closest earlier frame boundary, otherwise it is moved to the closest later frame boundary. The samples that are prepended/appended are chosen either randomly from the truncated options, or the samples are taken from a truncated option that is designated 'default' by the creator of the personalized message.

(b) Mathematically determine how much the last parts (q samples each) of the source options differ and how much the first parts (r samples each) of the target options differ. If the last parts of the source options are more similar than the first parts of the target options, the transition point is moved to the closest earlier frame boundary, otherwise to the closest later boundary. The samples that are prepended/appended can be chosen either randomly from one of the truncated options, a creator-assigned default option could be chosen, or a more complex algorithm, such as averaging the sample values over all truncated options, could be adopted.

Similarity of two sequences of samples can be determined using well-known mathematical algorithms that return a value between 0 (dissimilar) and 1 (identical). One example of a function that computes such a similarity is:

(Equation 1) 2*(SUMi:0<=i<N:s(i)*t(i))/(SUMi:0<=i<N:s(i)*s(i)+t(i)*t(i)),

where s and t are two sequences of sample values, each having a length of N samples.

It can easily be seen that this formula returns a value of 1 when all samples s(i) and t(i) are identical. The more the sample values differ, the closer to 0 this value will get.

The similarity of more than two sequences of samples can be determined by averaging the similarities of all possible pairs of sequences of samples.

Fig. 20 further illustrates this scenario. The last q samples 54 are different for each of the source options n-1 and n-2. The first r samples 56 are different for each of the target options n and n+1.

If the decision is made to move the transition point later in time, the result is depicted in Fig. 21. In this figure, the last q samples of each of the source options n-2 and n-1 are removed. One of the segments of q removed segments is prepended to each of the source options n and n+1.

If the decision is made to move the transition point earlier in time, the result is depicted in Fig. 22. In this figure, the first r samples of each of the target options n and n+1 are removed. One of the segments of r removed segments is appended to each of the source options n-1 and n-2.

No matter in what direction the audio transition point is moved, the last frames of the source options will not be identical. Depending on how much these last frames differ, and how much audio data is lost during the removal of audio samples, the transition will have inaudible, small or big artifacts during playout since the history buffer of the audio encoder for the target options cannot be initialized such that it is correct (seamless) for all possible transitions from source to target options.

It is very unlikely, that the ending of source fragments or the beginning of target fragments in a transition are very dissimilar, since this would mean that at least some transitions from source to target options will already have artifacts in the uncompressed case. This situation will likely be detected and repaired during creation of the personalized message. In the case that the last parts of the source options are reasonably similar, any of them could be used to provide the frame to be used for initializing the history of the audio encoder without leading to audible artifacts.

An example in which the last parts of options that are intended to be identical can actually differ is when the source options are captured from an analog tape. In this case, some sampling errors/jitter will occur during the capture process, leading to slight dissimilarities. However, these small differences will generally not lead to audible artifacts later on in the process.

After each transition in the template has been processed according to the mechanism just disclosed, either manually or mechanically, the last options in the template, i.e., those that have NO successors, and are the last ones that will be played, will be padded with silence (zero sample values) to make their length also an exact multiple of the frame length of the intended compression scheme. This to ensure that the audio encoder will not discard the last remaining part of each such last option because it is no complete frame in length.

Encoding

Assuming that alignment has been completed for all transitions/options in the personalized message, the last step part of this invention is actual encoding (compression) of the individual options.

The main difference between ordinary encoding of a standalone audio fragment and the encoding of an option in a personalized message is that options that can be played directly before that option to be encoded must be taken into account. It is necessary to encode at least one frame of the previous material before the actual fragment to be encoded to build the history of the psycho-acoustic block in a perceptual encoder such as one based on the MPEG and AC-3 compression standards. By building up history, the transition between options can be made perfectly seamless.

In case the option to be encoded is not preceded by a transition, i.e., has no options that can be played before it (because it will be always be played first in the personalized message), it is encoded as is, without needing any special processing.

If the option to be encoded is a target option in a transition, the last frame of any of the source options of that transition is temporarily prepended to the target option to be encoded. The resulting target option (prepended with one frame) is encoded. After encoding, the first frame is stripped-off from the encoded result. As mentioned, this first frame purely serves to build-up a history in the encoder to thus enable seamless transitions from any of the possible source options to this target option. The choice for the frame to be prepended is presented here as arbitrary, since alignment has already ensured that the last frame of each source option in a transition is identical to the last frame of any other of the source options, wherever possible. In the cases that this could not be accomplished (in certain 'junction' transitions), either a random choice can be made, a certain option that is marked as 'default' (e.g., by the creator of the personalized message) will be selected, or any other selection algorithm can be used.

Stripping a single frame from the beginning of a compressed audio file usually is a very simple algorithm. For example, in MPEG Layer II or AC-3 compressed audio, frames can be added and removed independently without invalidating the file. Also, each frame starts with a defined (sync) code that also contains the size of the frame, so the start of the next frame can be found easily.

As illustration, encoding for the various transition cases (branch, confluence, junction) is shown in a number of Figures.

Fig. 6 shows how encoding of target options takes place in a branch transition. The last frame 58 of the (single) source option is temporarily copied in front of each target option before encoding.

Fig. 10 shows how encoding of target options takes place in a confluence transition. Any of the (identical) last frames 60 of the source options is taken and temporarily copied in front of each target option before encoding

Fig. 13 shows how encoding of target options takes place in scenario 1 of a junction transition. Any of the (identical) last frames 54+56 of the source options is taken and temporarily copied in front of each target option before encoding.

Fig. 16 shows how encoding of target options takes place in scenario 2 of a junction transition. One of the (only partly identical) last frames 54+56 of the source options is taken and temporarily copied in front of each target option before encoding.

Fig. 19 shows how encoding of target options takes place in scenario 3 of a junction transition. One of the (different) last frames 62 of the source options is taken and temporarily copied in front of each target option before encoding.

It can easily be seen that the encoding of the target options in scenario 4 of a junction transition is very similar to that of Fig. 19: One of the (different) last frames 62 of the source options (see also Fig. 21 and Fig. 22) is taken and temporarily copied in front of each target option before encoding.

Although the invention has been shown and described with respect to illustrative embodiments thereof, various other changes, omissions and additions in the form and detail thereof may be made therein without departing from the spirit and scope of the invention. It can easily be seen by someone moderately skilled in the art that the invention can be applied in any domain where separate audio fragments must be compressed and concatenated or selected later. Domains include DVD, Digital television, Internet streaming media, and many others.

What is claimed is:

CLAIMS

1. A method of preparing a plurality of digital audio fragments to allow switching between at least one source fragment and at least one target fragment said method comprising:

aligning an end of said at least one source fragment with a beginning of said at least one target fragment, for all possible valid combinations of said at least one source fragment and said at least one target fragment; wherein said at least one source fragment is aligned to be a length that is an exact multiple of a predetermined number.

2. The method of claim 1 further comprising:

after said step of aligning an end of said at least one source fragment; moving a sequence of audio samples from a digital audio fragment which was shortened because of said alignment step, to a plurality of digital audio fragments which were lengthened because of said alignment step.

3. The method of claim 2 wherein said moved sequence of audio samples is a length which will result in said at least one source fragment to be a length that is an exact multiple of said predetermined number

4. The method of claim 1 further comprising:

after said step of aligning an end of said at least one source fragment; moving a sequence of audio samples from said end of said at least one source fragment to said beginning of said at least one target fragment; wherein said sequence of audio samples is a length which will shorten said at least one source fragment to be a length that is an exact multiple of said predetermined number.

5. The method of claim 1 further comprising:

after said step of aligning an end of said at least one source fragment; moving a sequence of audio samples from said beginning of said at least one target fragment to said end of said at least one source fragment wherein said sequence of audio samples is a length which will lengthen said at least one source fragment to be a length that is an exact multiple of said predetermined number.

6. The method of claim 5 further including:

moving a second sequence of audio samples from said beginning of said at least one target fragment to said end of said at least one source fragment wherein said second sequence of audio samples is a length equal to said predetermined number.

7. The method of claim 1, wherein the digital audio fragments are compressed as a sequence of frames, wherein each frame comprises a sequence of audio samples; and wherein a length of said frame is said predetermined number.

8. The method of claim 7, wherein said compression scheme includes encoding a sequence of said frames wherein a subsequent frame encoding is dependent upon an encoding of at least one preceding frame

9. The method of claim 7, wherein ends of a plurality of source fragments are aligned with beginnings of a plurality of associated target fragments by moving identical audio samples from the beginning of said target fragments to the end of said source fragments so that a resulting end of said source fragments aligns at an exact multiple of said predetermined number, and a resulting last frame of said plurality of source fragments is identical.

10. The method of claim 7, wherein ends of a plurality of source fragments are aligned with beginnings of a plurality of associated target fragments by moving identical audio samples from the beginning of said target fragments to the end of said source fragments; so that the end of said source fragments align at an exact multiple of said predetermined number, and wherein at least one audio sample is identical at the end of all of said source fragments, however the last full frame of audio samples are not identical for all said source fragments.

11. The method of claim 7, wherein the ends of a plurality of source fragments are aligned with beginnings of a plurality of associated target fragments by moving identical audio samples from the end of said source fragments to the beginning of said target fragments; so that the resulting end of said source fragments aligns at an exact multiple of said predetermined number, wherein at least one more audio sample is identical at the beginning

of all of said target fragments, however the first full frame of audio samples is not identical for all said target fragments.

12. The method of claim 7, wherein the ends of a plurality of source fragments are aligned with the beginnings of a plurality of associated target fragments by moving audio samples from the beginning of a first target fragment to the end of said plurality of source fragments; and removing an identical number of audio samples from the beginning of said remaining plurality of fragments, so that the resulting end of said source fragments aligns at an exact multiple of said predetermined number.

13. The method of claim 7, wherein the ends of a plurality of source fragments are aligned with the beginnings of a plurality of associated target fragments by moving a number of samples from the end of a first source fragment to the beginning of said plurality of target fragments; and removing an identical number of audio samples from the end of said remaining source fragments, so that the resulting end of said source fragments aligns at an exact multiple of said predetermined number.

14. The method of claim 7, wherein at least one digital audio fragment has an end time that is later than the beginning time of any other digital audio fragments, wherein said at least one digital audio fragment is aligned at an exact multiple of said predetermined number by adding empty audio samples to the end of said at least one audio fragment.

15. The method of claim 8, further including:

copying a last frame of a source fragment to the beginning of at least one target fragment;

compressing said at least one target fragment using said compression scheme;

removing data from the beginning of said compressed at least one target fragment, said data corresponding to a first frame of said at least one target fragment.

16. A system for preparing a plurality of digital audio fragments for transmission to allow a switching device to switch between at least one source fragment and at least one target fragment; said system comprising:

an audio aligner module, coupled to a source of said plurality of audio fragments, to align beginning and ends of said plurality of audio fragments to selected times based on an exact multiple of a predetermined number;

an audio compression module, coupled to said audio aligner module, to compress said plurality of audio fragments as a sequence of frames, wherein each frame comprises a sequence of audio samples; and wherein a length of said frame is said predetermined number.

17. The system of claim 16, wherein said plurality of audio fragments are transmitted using a transport mechanism selected from one of MPEG compliant, digital television, dvd broadcast, dvd storage, CD ROM, and internet.

18. The system of claim 16, wherein said plurality of audio fragments are compressed using AC-3

19. The system of claim 16, wherein said plurality of audio fragments are compressed using MPEG Layer II

20. A switching apparatus, to switch between a plurality of audio fragments, wherein said audio fragments are prepared so that so that an end of said at least one source fragment is aligned with a beginning of said at least one target fragment and wherein said least one source fragment is aligned to be a length that is an exact multiple of a predetermined number; wherein said switch apparatus switches between at least one source fragment and at least one target fragment at said alignment.

21. The switching apparatus of Claim 20 wherein said switching apparatus receives said plurality of audio fragments transmitted using a transport mechanism selected from one of MPEG compliant, digital television, dvd broadcast, dvd storage, CD ROM, and internet. a memory module, said memory module to receive at least one target fragment at a time before switching to said at least one target fragment.

22. The switching apparatus of claim 20 wherein said switching apparatus is a receiver for MPEG encoded media streams.

23. The switching apparatus of claim 20 wherein said switching apparatus selected from one of set top box, dvd player, personal computer, digital television set, video server, and video on demand server.

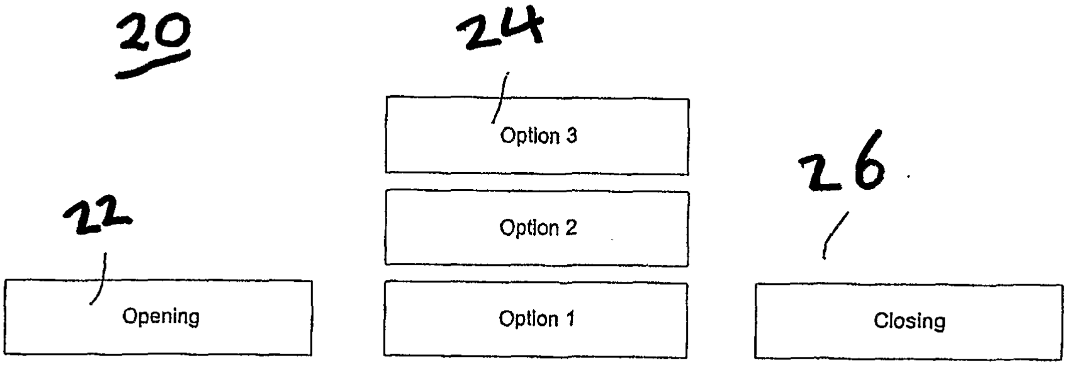


Fig. 1: A very simple personalized message structure

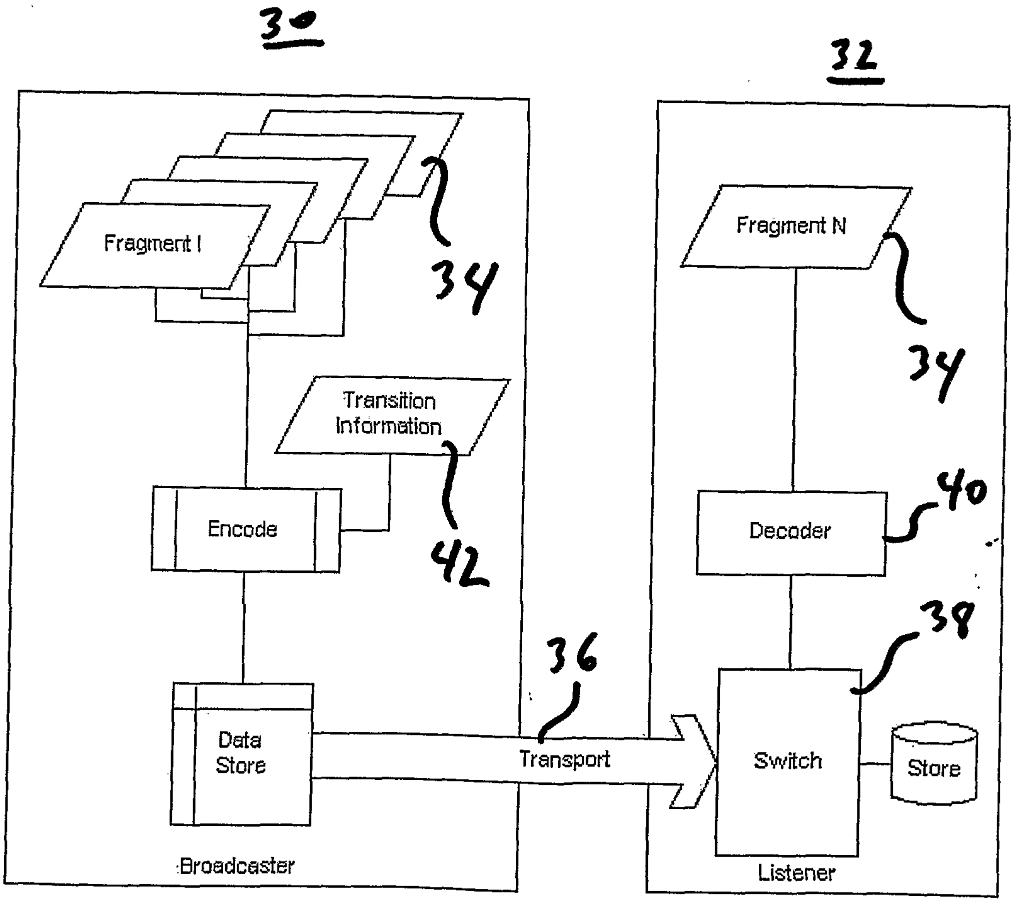


Fig. 2: Processing Model

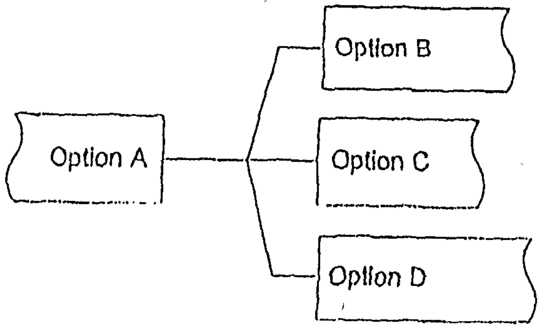


Fig. 3A: Branch (1 source option; N>1 target options)

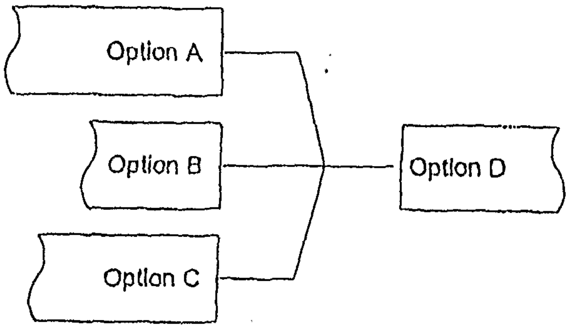


Fig. 3B: Confluence (M>1 source options; 1 target option)

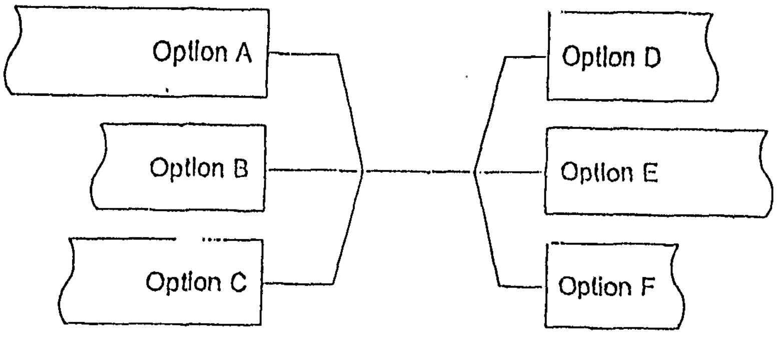


Fig. 3C: Junction (M>1 source options; N>1 target options)

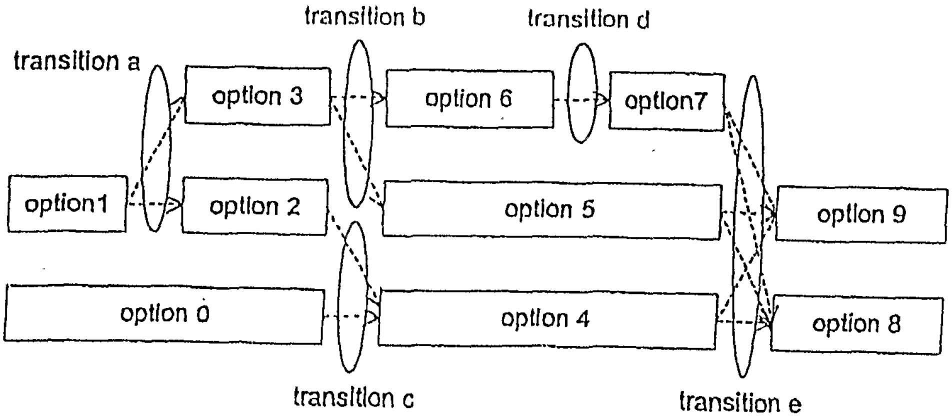


Fig. 3D: Personalized message with 5 transitions

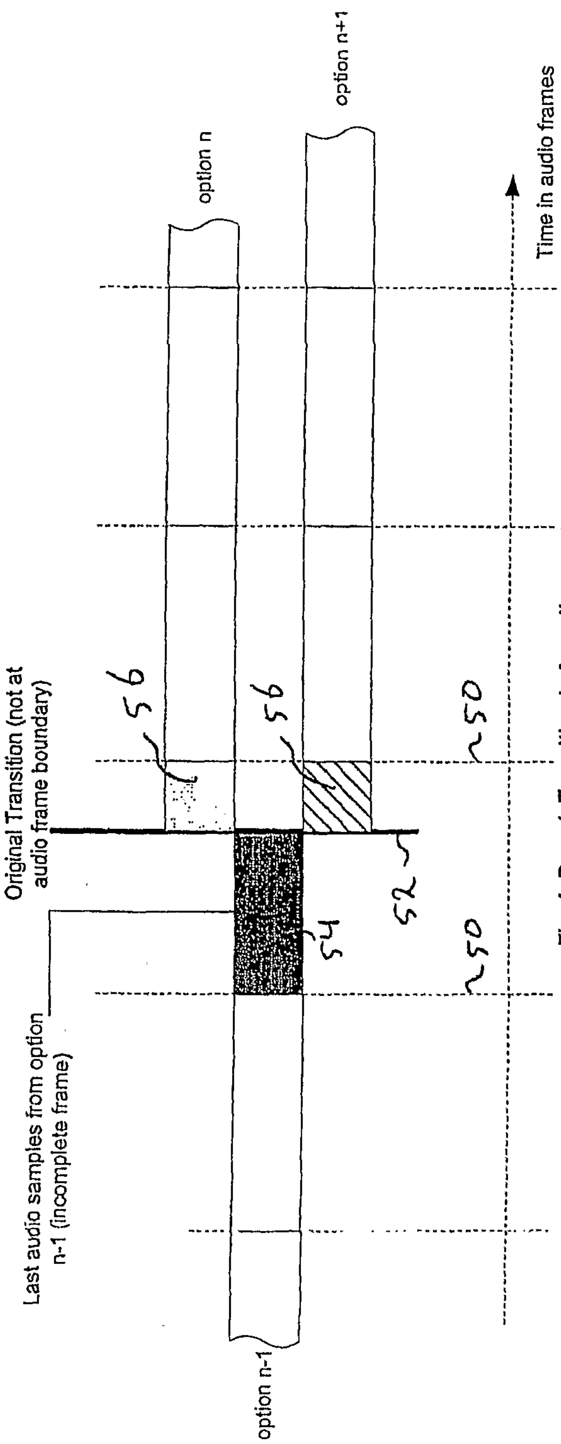

option n
option n+1
Time in audio frames
Original Transition (not at audio frame boundary)
Last audio samples from option n-1 (incomplete frame)
option n-1
56
56
50
50
52
54

Fig. 4: Branch Transition before alignment

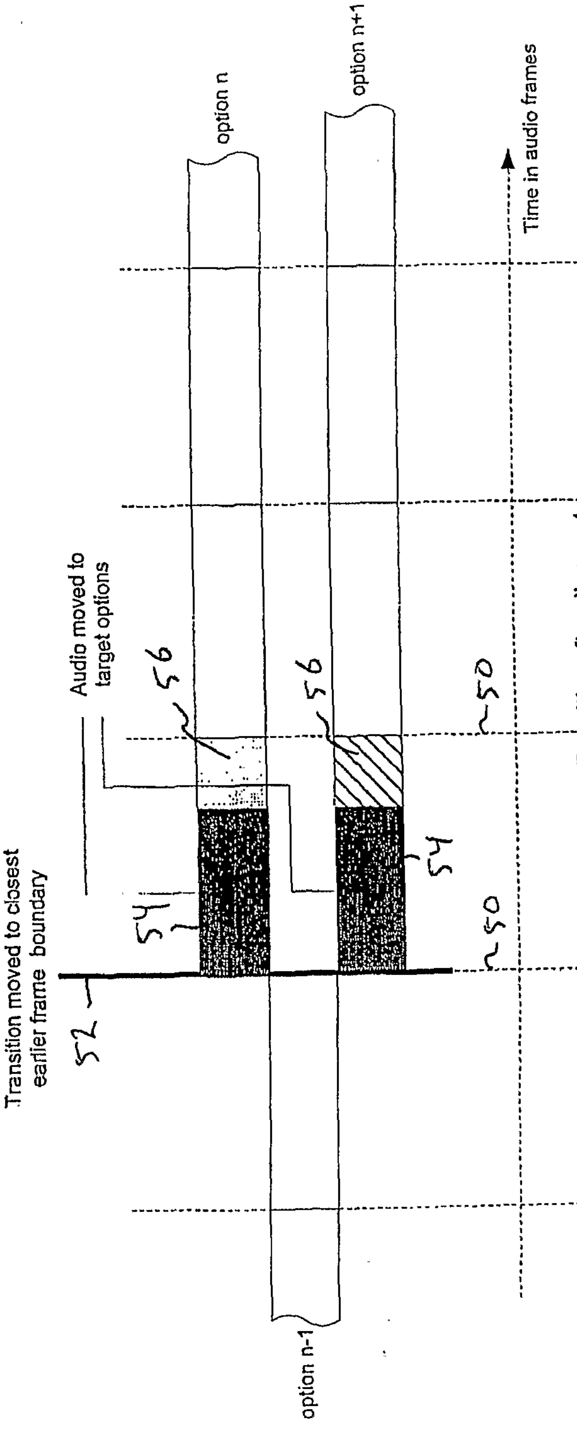

Transition moved to closest earlier frame boundary
52
Audio moved to target options
56
56
54
54
50
50
option n
option n+1
option n-1
Time in audio frames

Fig. 5: Branch Transition after alignment

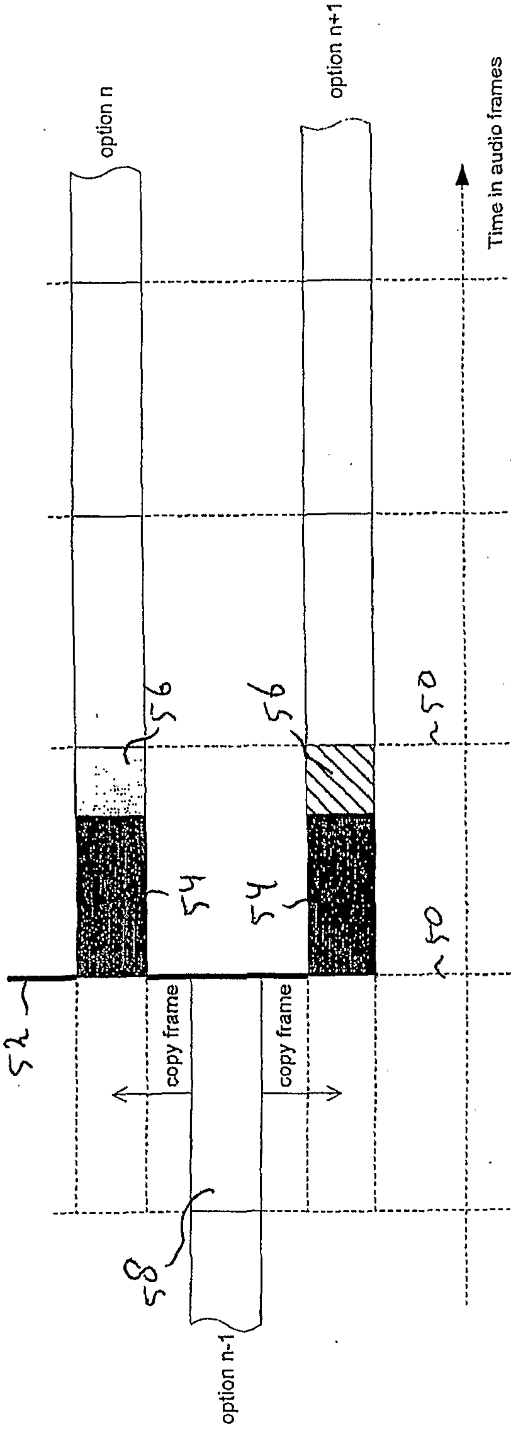


Fig. 6: Encoding target options in Branch Transition

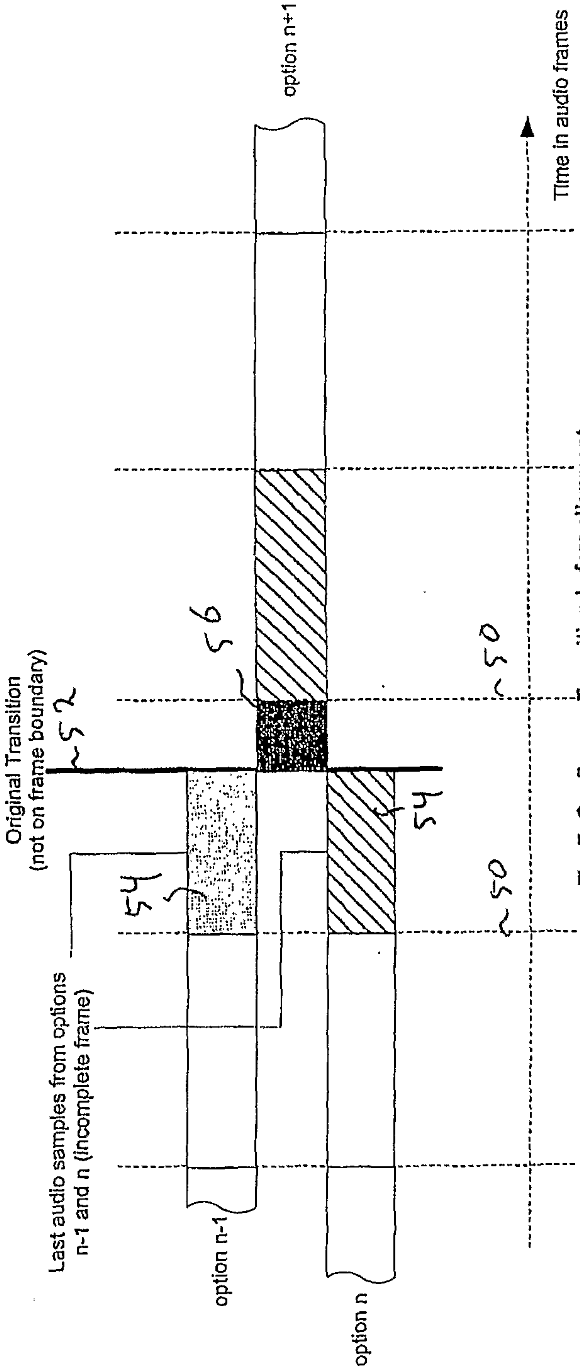


Fig. 7: Confluence Transition before alignment

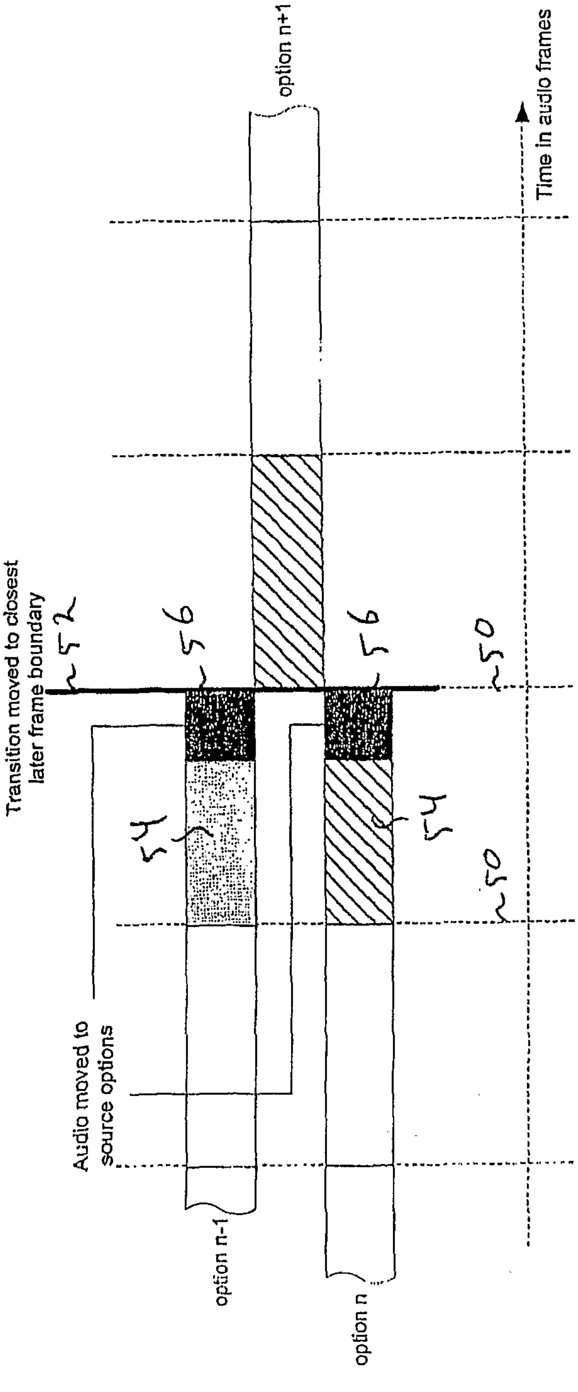

Transition moved to closest later frame boundary
52
56
56
50
50
54
54
Audio moved to source options
option n-1
option n
option n+1
Time in audio frames

Fig. 8: Confluence Transition after alignment (step 1)

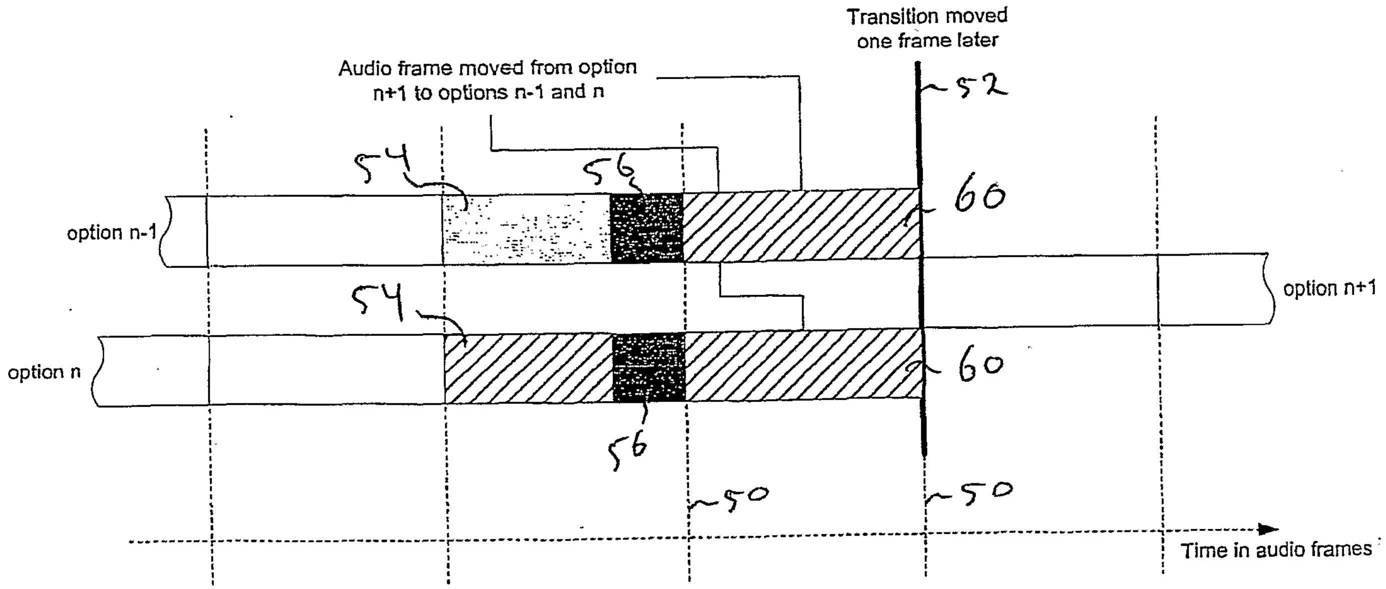


Fig. 9: Confluence Transition after alignment (step 2)

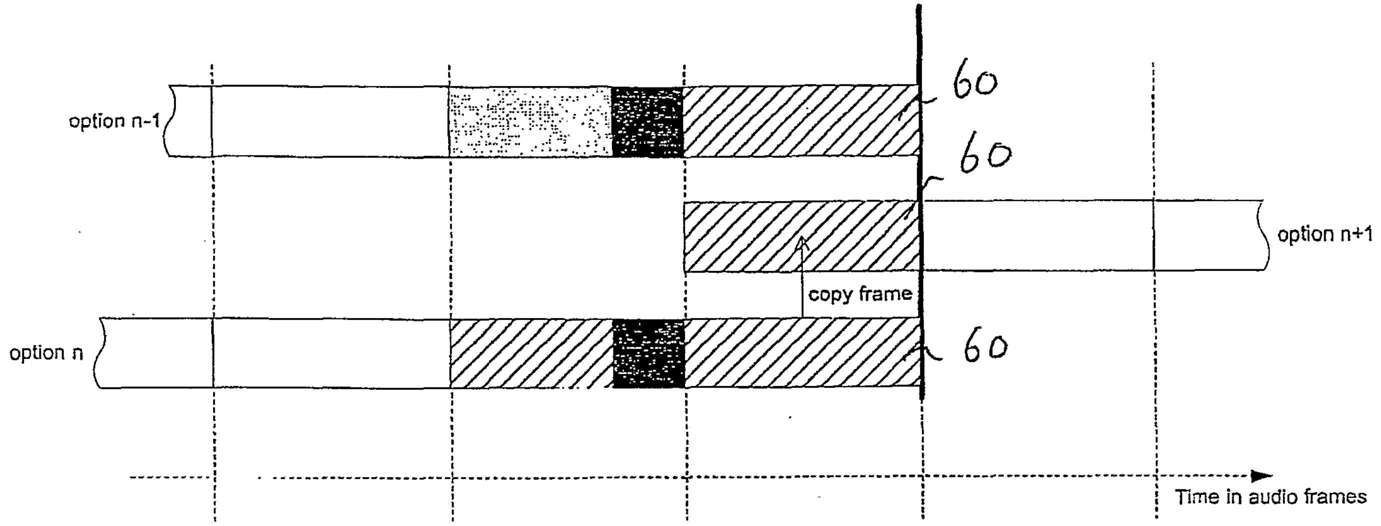


Fig. 10: Encoding target options in Confluence transition

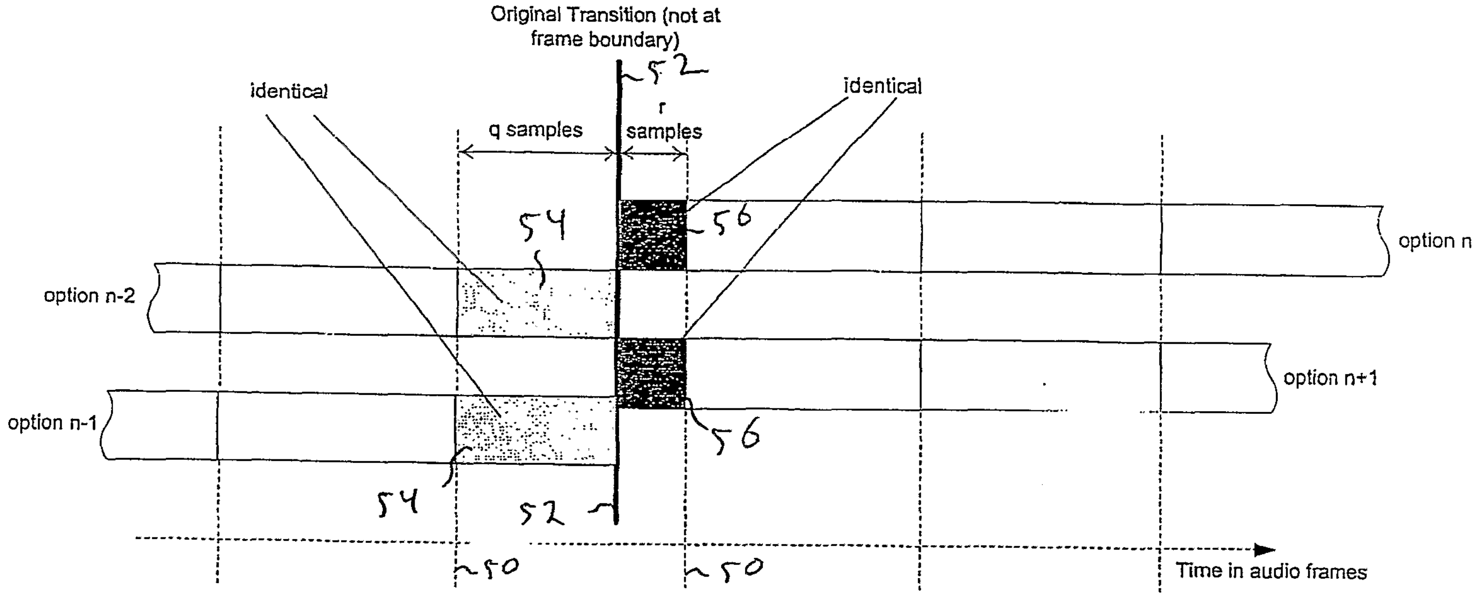


Fig. 11: Junction Transition - Scenario 1 before alignment

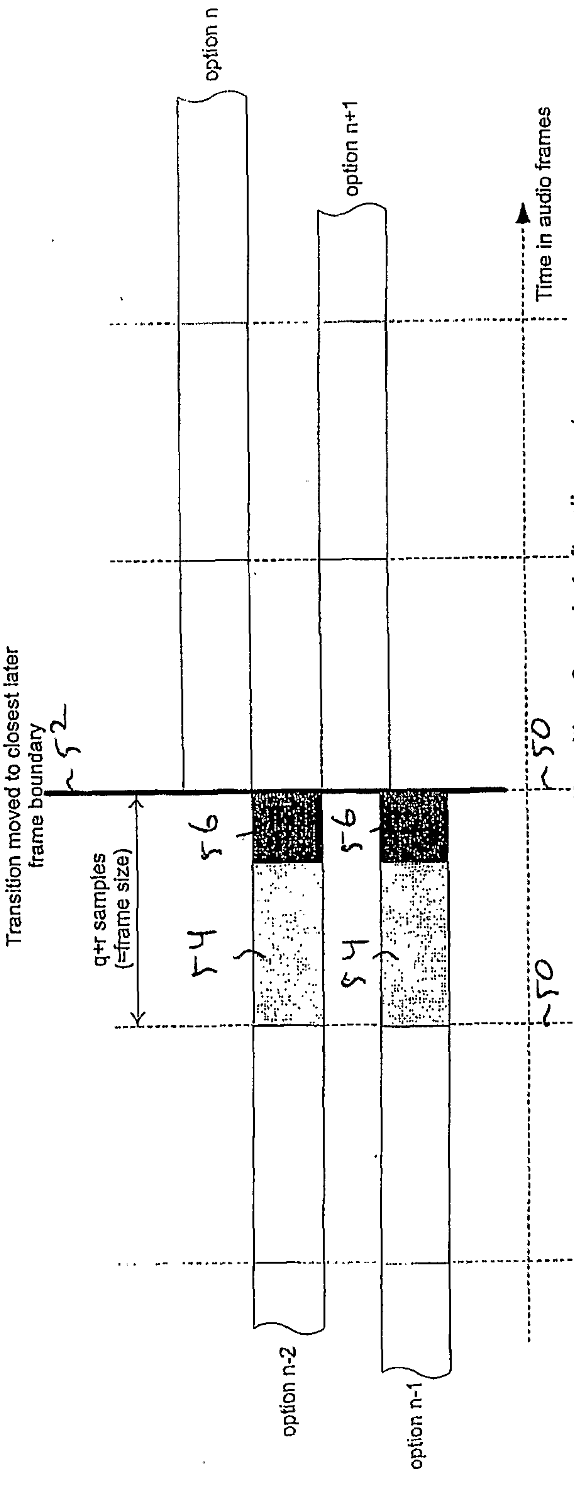


Fig. 12: Junction Transition - Scenario 1 after alignment

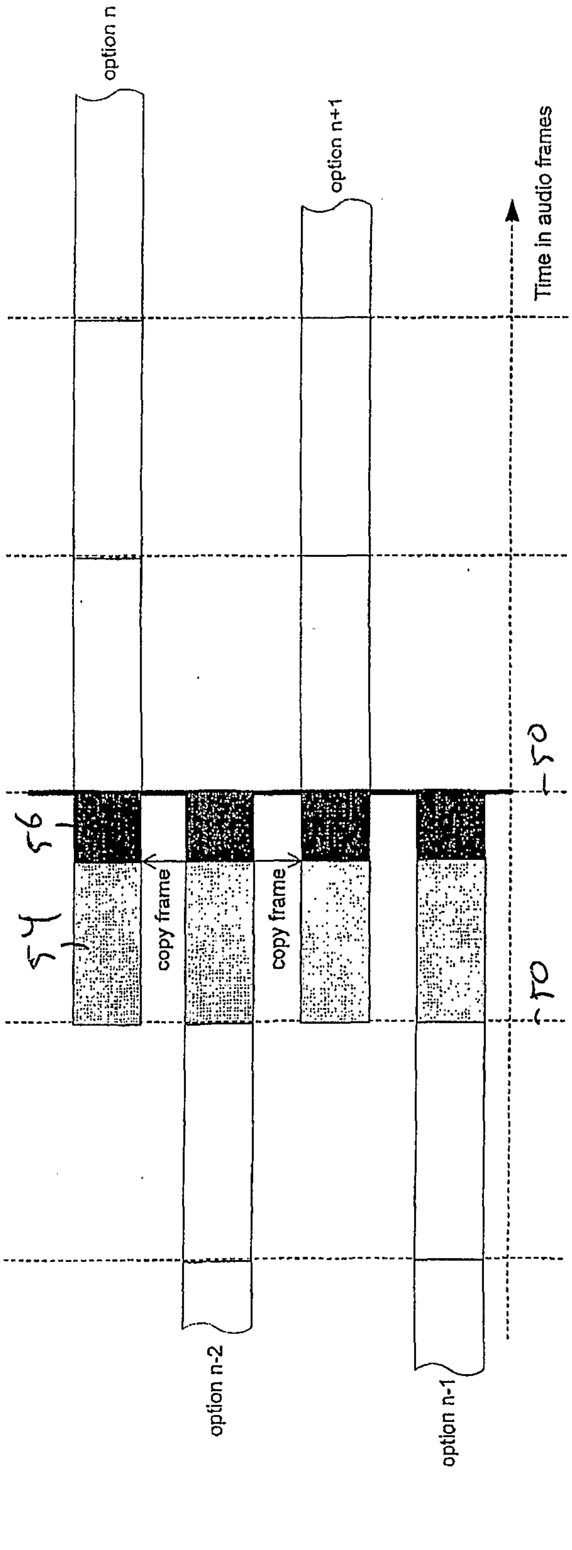


Fig. 13: Encoding target options in Junction Transition - Scenario 1

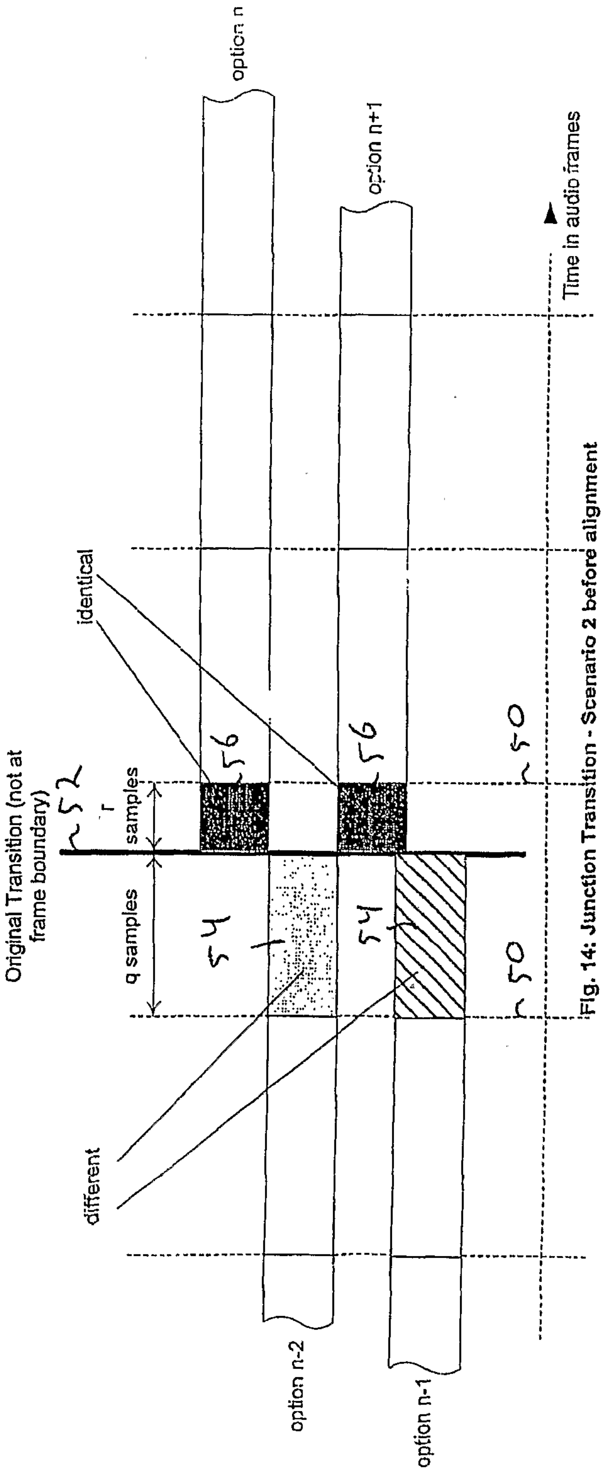


Fig. 14: Junction Transition - Scenario 2 before alignment

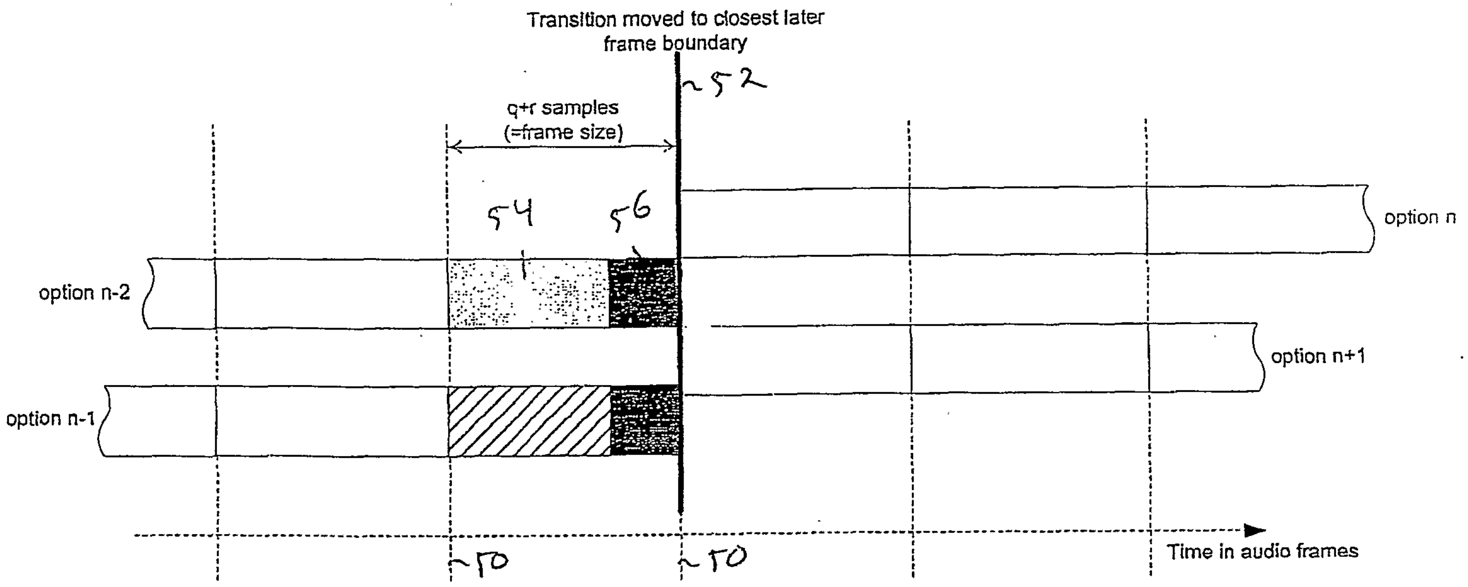


Fig. 15: Junction Transition - Scenario 2 after alignment

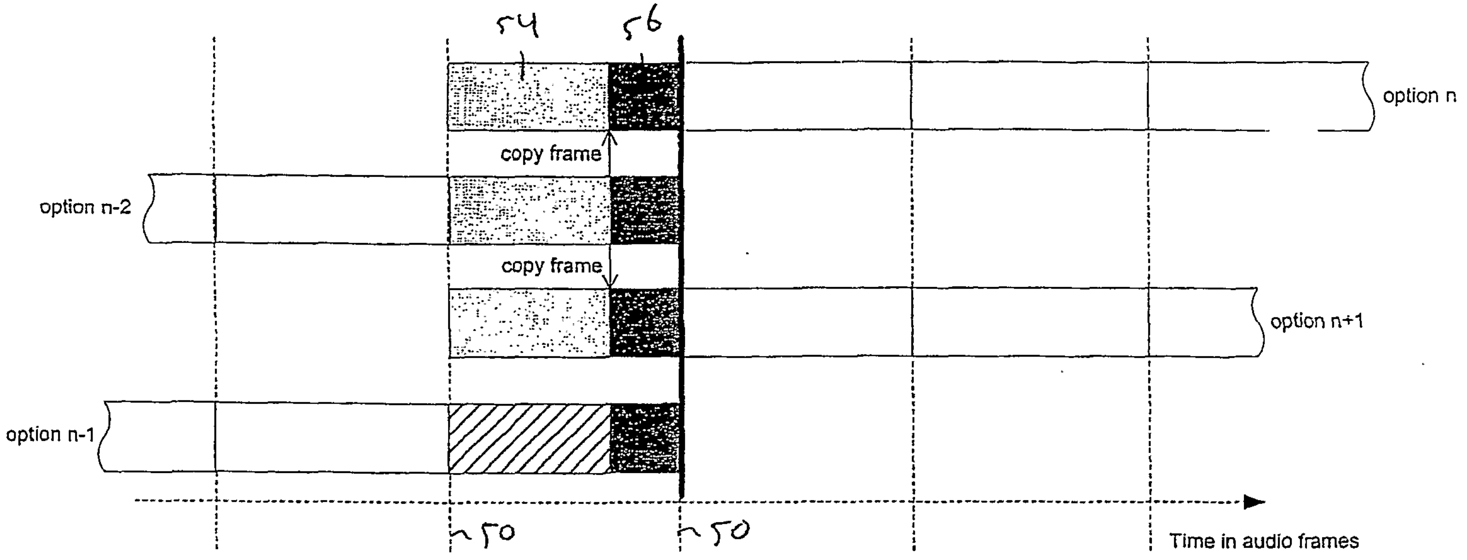


Fig. 16: Encoding target options in Junction Transition - Scenario 2

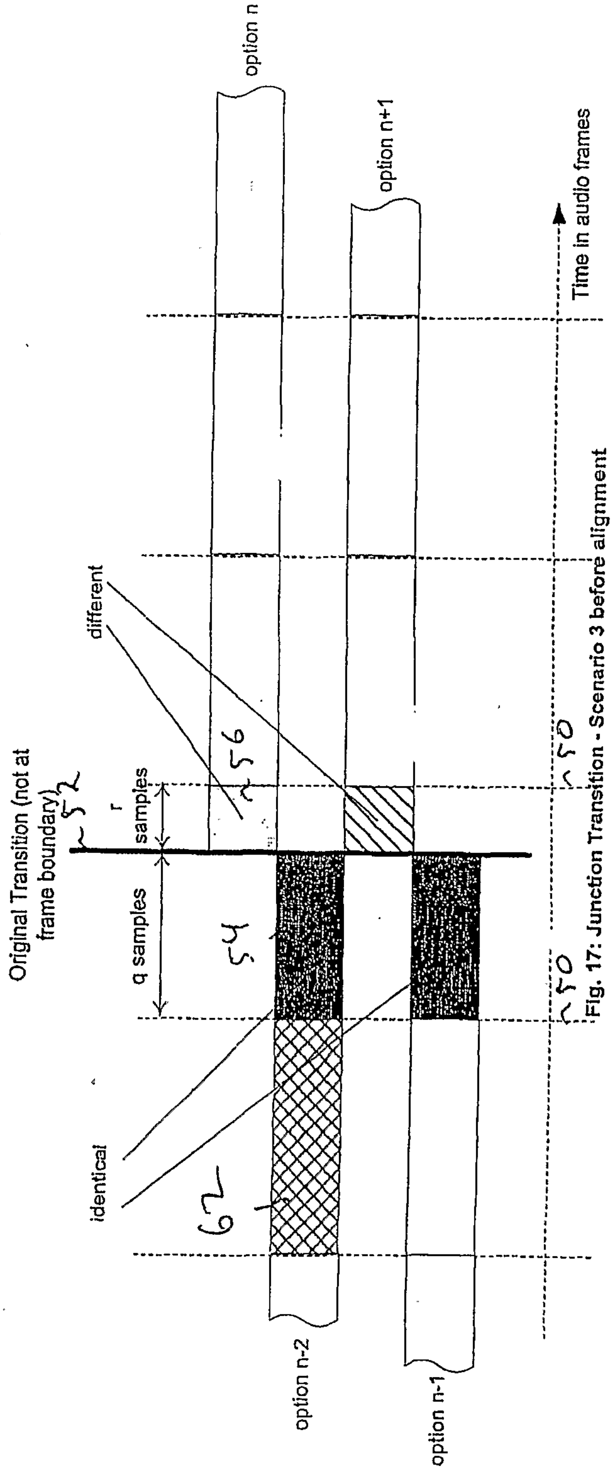


Fig. 17: Junction Transition - Scenario 3 before alignment

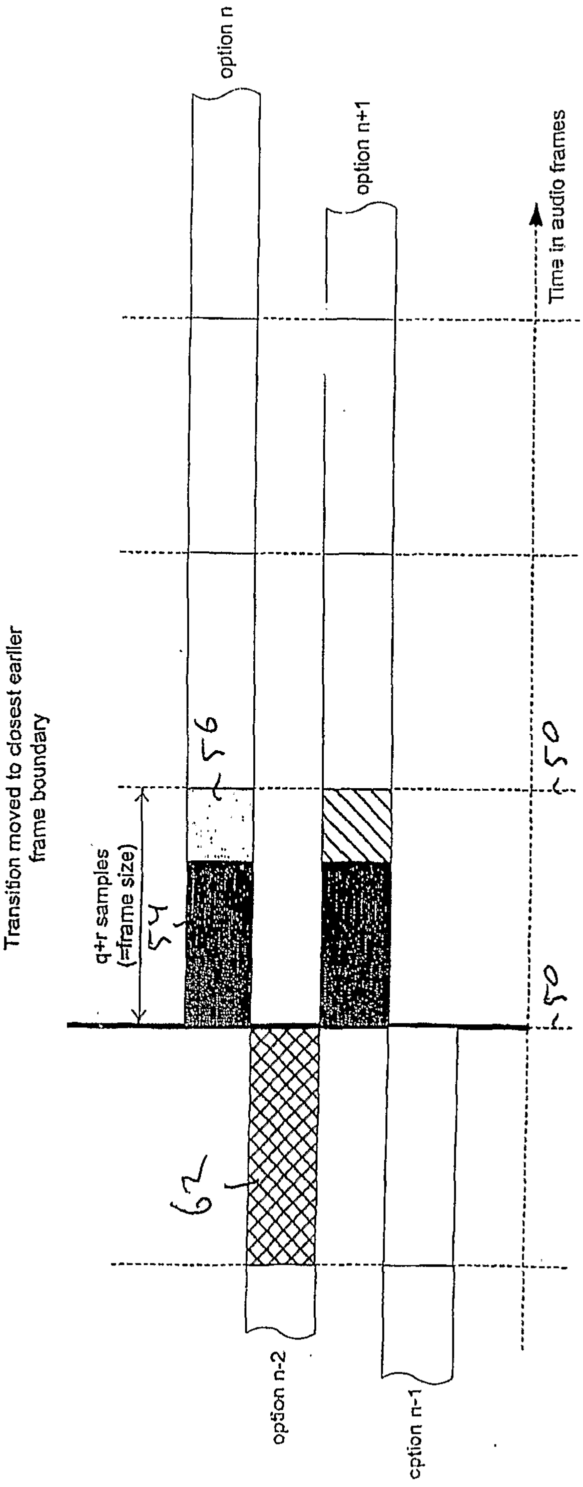

Transition moved to closest earlier frame boundary
q+r samples (=frame size)
54
56
50
50
62
option n
option n+1
option n-2
option n-1
Time in audio frames

Fig. 18: Junction Transition - Scenario 3 after alignment

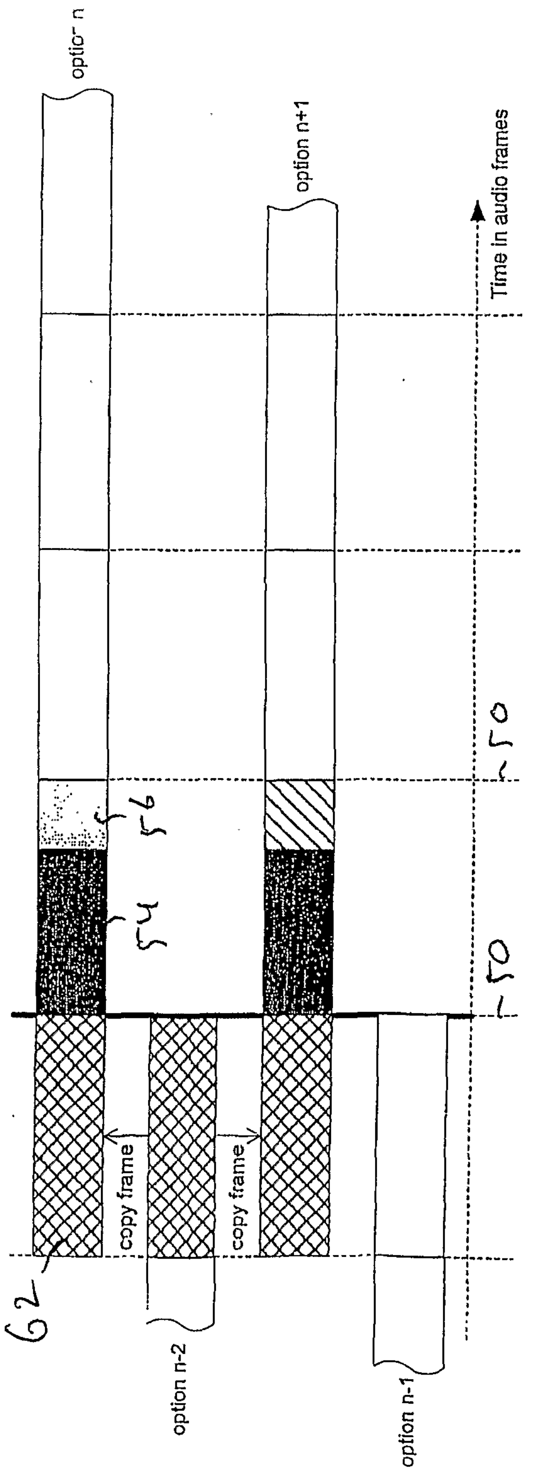


Fig. 19: Encoding target options in Junction Transition - Scenario 3

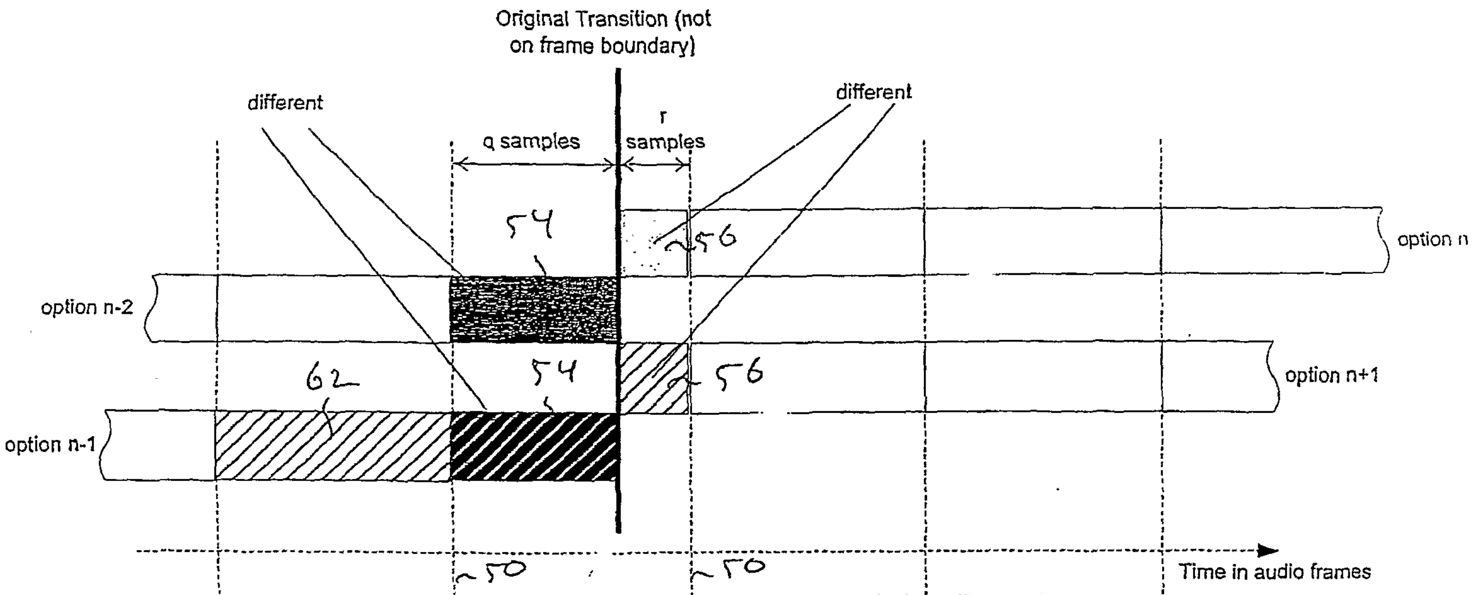


Fig. 20: Junction Transition - Scenario 4 before alignment

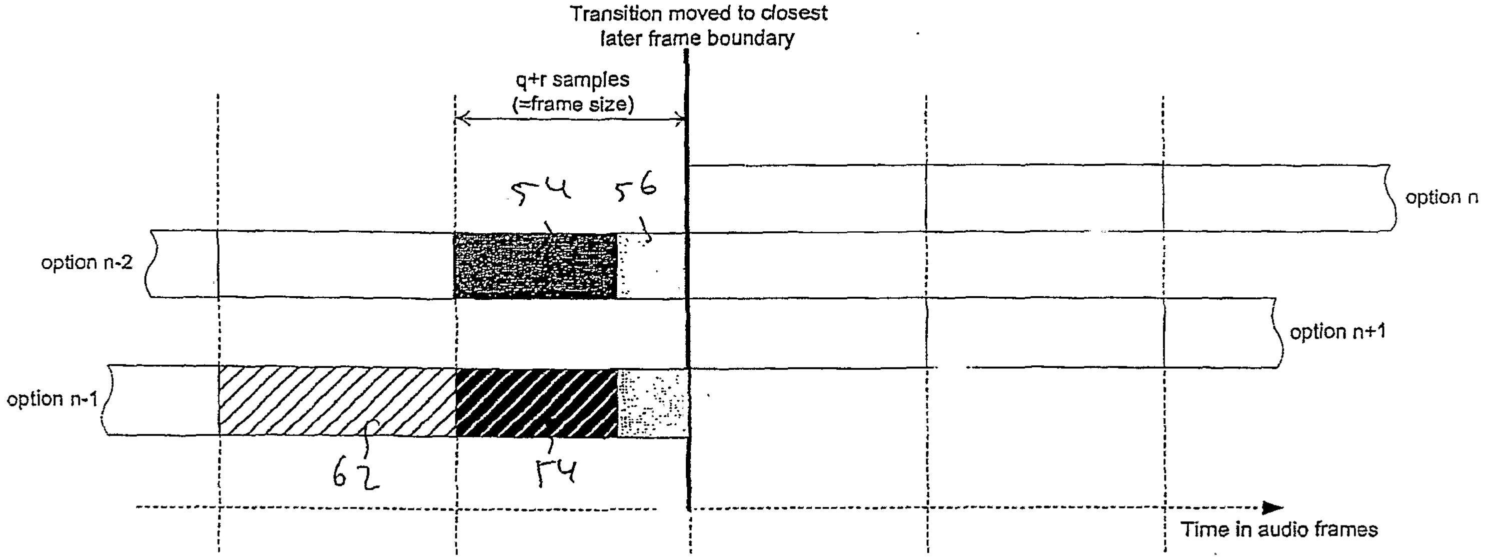

Transition moved to closest later frame boundary
q+r samples (=frame size)
54
56
option n
option n-2
option n+1
option n-1
62
54
Time in audio frames

Fig. 21: Junction Transition - Scenario 4 after aligning (move later)

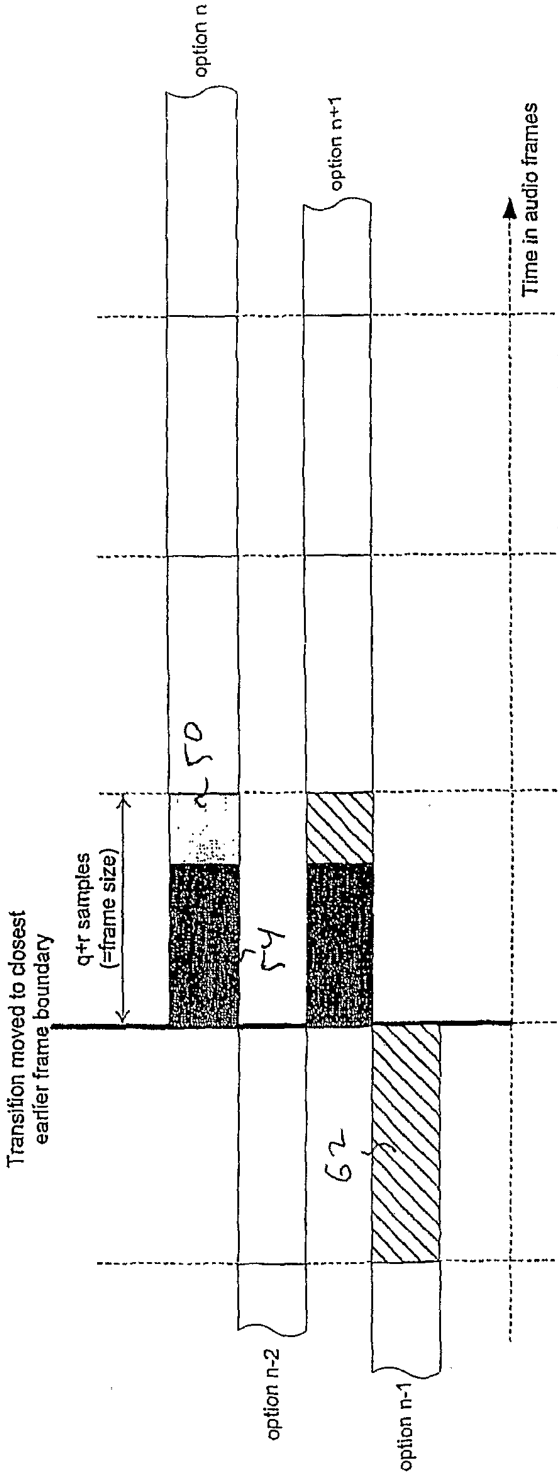

Transition moved to closest earlier frame boundary
q+r samples (=frame size)
option n
option n+1
option n-2
option n-1
Time in audio frames
50
54
62

Fig. 22: Junction Transition - Scenario 4 after aligning (move earlier)