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

(11) EP 3 327 723 A1

(12) EUROPEAN PATENT APPLICATION

(43) Date of publication:
30.05.2018 Bulletin 2018/22

(51) Int Cl.:
G10L 21/04 *(2013.01)*

(21) Application number: **16306550.1**

(22) Date of filing: **24.11.2016**

(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 MK MT NL NO PL PT RO RS SE SI SK SM TR
Designated Extension States:
BA ME
Designated Validation States:
MA MD

(71) Applicant: **Listen Up Technologies Ltd**
4330306 Raanana (IL)

(72) Inventors:
- **LEVI, Aharon Roni**
  NESS TZIONA (IL)
- **PETKOVIKJ, Martin**
  KUMANOVO (MK)
- **GERAZOV, Branislav**
  1000 SKOPJE (MK)
- **SERRA, Yves Joseph Michel**
  92130 ISSY LES MOULINEAUX (FR)
- **OFFER, Ronen**
  ST RAANANA (IL)
- **MIZRAHI, Ronen**
  TENAFLY, NJ 07670 (US)
- **SIMEVSKI, Igor**
  SKOPJE (MK)

(74) Representative: **Regimbeau**
87 rue de Sèze
69477 Lyon Cedex 06 (FR)

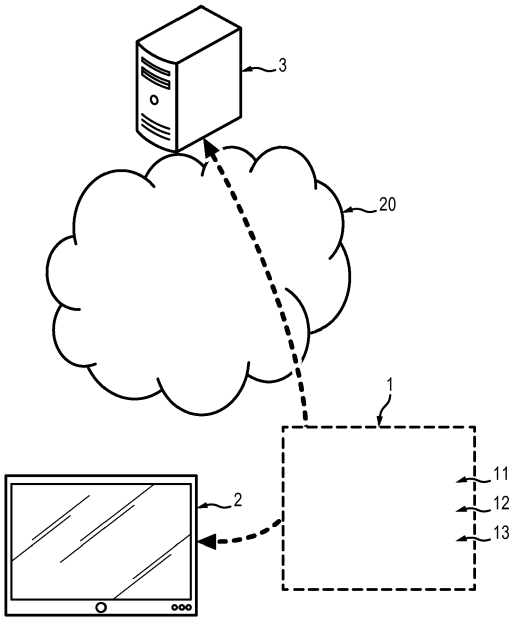
(54) METHOD FOR SLOWING DOWN A SPEECH IN AN INPUT MEDIA CONTENT

(57) The present invention relates to a method for slowing down speech in an input audio signal constituted by a sequence of audio frames, comprising performing steps of:
(a) classifying the audio frames as speech, non-speech, or pause, so as to divide said audio signal into speech segments bounded by non-speech segments;
(b) for each speech segment:
1. dividing it into a sequence of intervowel segments;
2. calculating an average intervowel distance (T_{avg}), and determining a non-linear stretching transfer function mapping an input intervowel distance (T_{in}) to an output intervowel distance (T_{out}) as a function of said average intervowel distance (T_{avg}) and a given target intervowel distance (T_{target});
3. for each intervowel segment, stretching it using the determined stretching transfer function so as to generate updated audio frames;

(c) generating as output signal the input audio signal wherein for each intervowel segment of each speech segment the corresponding audio frames have been replaced by the updated audio frames.

The present invention also relates to an equipment for carrying out said method.

FIG. 1



EP 3 327 723 A1

Description

FIELD OF THE INVENTION

[0001] The field of this invention is that of audio signal processing.

[0002] More precisely, the invention relates to a method for slowing down speech in media content.

BACKGROUND OF THE INVENTION

[0003] Audio-visual media are omnipresent nowadays, and a large part of the information is provided in the form of speech, often with a high speaking rate for faster distribution.

[0004] This makes the flow of information sometimes difficult to grasp, especially for the non-native speakers, the elderly, or the hearing-impaired.

[0005] There is consequently an immediate demand for a way to increase the intelligibility of fast speech, in particular by slowing it down to speeds that are more comfortable for the listener.

[0006] However, it is not possible to simply play an audio signal containing speech at a slower pace, because it also results in an alteration of the pitch and warping of the spectrum. Consequently the speech intelligibility is not increased, but even decreased.

[0007] Thus, there have been proposed approaches towards "Time Scale Modification" (TSM), i.e. slowing down or speeding up the audio signal without affecting the frequency content, such as the perceived pitch of any tonal components.

[0008] TSM may be performed by processing the audio signal directly in the time domain, or in a transformation domain, e.g. in the Fourier domain.

[0009] In particular, "SOLA" (synchronized overlap-add) is a type of Time-Domain TSM algorithms, which perform slow-down by repeating segments locally. They determine the overlap points by finding the maximum cross-correlation in order to minimize discontinuity distortion, but such distortion is always there, though typically attenuated by a cross-fade on the overlap add. These methods are especially well-suited to speech.

[0010] For example, in the patent document US7853447, a maximum is found in every search range defined with the pitch period and speech is divided according to these maximums. TSM is done to these segments with linear cross-fading. The processed sections of the speech are inserted in between not processed sections to slow down speech. The processed sections replace the not processed sections to speed up the speech.

[0011] In the patent document US6484137, the output signals are composed according to a table, which contains a pattern of what segments to process and what segments to pass through. More specifically, low-energy segments, segments with low probability of containing a human voice, high-stationarity segments, and/or segments with no detected distortion are selected with priority. Linear cross-fading is used to make the data compression/expansion. The correlation is calculated for the subband that contains the highest energy and the subband that contains the pitch frequency.

[0012] The patent document US7412379 proposes a dual approach in which unvoiced frames are expanded using a parametric technique and voiced frames are expanded using a waveform based technique, such as SOLA. The unvoiced frames are expanded by inserting noise colored using linear predictive coefficients extracted from the speech signal.

[0013] In practice, such methods appear to not have an important effect on intelligibility, as the speech is not slown down in a "natural way".

[0014] In the document "Methods of Improving Speech Intelligibility for Listeners with Hearing Resolution Deficit", by A. Kupryjanow and A. Czyzewski 2012, a new method for real-time time scale modification of speech signal is proposed in which the syllable rate is estimated, and a stretching coefficient for every signal frame is calculated based on the syllable rate. This method does increase intelligibility, but it stretches the audio frame by frame, only using the speech rate within 1.5 s as context and the phonetic character of the frame. Thus it fails to capture the local dynamics in the speech rate as evident by the variation of the distances between succeeding vowels. Specifically, it does not put special focus on the fast segments in an utterance, which impact intelligibility the most. On the frame level, the method also does not strategically spread the application of the stretching algorithm across the parts favorable for stretching to reduce processing artifacts, instead applying it on the fly for each frame as it comes.

[0015] Consequently, there is a still a need for a new method for slowing down speech designed to generate a high-quality natural sounding speech output, leading to improved intelligibility and consumer comfort.

SUMMARY OF THE INVENTION

[0016] For these purposes, the present invention provides according to a first aspect a method for slowing down speech in an input media content received by an equipment comprising a processing unit, the input media content comprising an input audio signal constituted by a sequence of audio frames, the method being characterized in that it comprises performing by the processing unit steps of:

(a) classifying the audio frames as speech, non-speech, or pause, so as to divide said audio signal of the media content into speech segments bounded by non-speech segments;

(b) for each speech segment:

1. dividing the speech segment into a sequence of intervowel segments;
2. calculating an average intervowel distance of

the speech segment, and determining a non-linear stretching transfer function mapping an input intervowel distance to an output intervowel distance as a function of said average intervowel distance and a given target intervowel distance;
3. for each intervowel segment of the speech segment, stretching the intervowel segment using the determined stretching transfer function so as to generate updated audio frames of the intervowel segment;

(c) generating an output media content comprising as output signal the input audio signal wherein for each intervowel segment of each speech segment the corresponding audio frames have been replaced by the updated audio frames.

[0017] Preferred but non limiting features of the present invention are as follow:

- step (a) comprises for each audio frame:
  - determining whether the audio frame is silent by analysing the energy of the audio signal within the frame;
  - if the audio frame is not determined as silent, classifying it as speech or non-speech;
- step (a) further comprises:
  - if the audio frame is determined as silent, classifying it as pause if it is a part of a procession of a plurality of silent audio frames, else classifying it as speech or non-speech according to neighbouring classified audio frames;
- said input media content is received as a stream, audio frames being successively read from an input buffer of a memory unit of the equipment to be classified, wherein:
  - Step (b) further comprises forwarding the generated updated audio frames to an output buffer;
  - step (a) further comprises:
    - ◦ if the presently read audio frame is classified as speech or pause and if the previously read audio frame is identically classified, regarding it as proper and forwarding it to a speech or pause segment buffer of the processing unit;
    - ◦ else if the presently read audio frame is classified as speech or pause and if the previously read audio frame is not identically classified, regarding it as not proper and forwarding it to an internal speech or pause segment buffer in the processing unit, after having performed step (b) on the contents of the speech or pause segment buffer as a segment and emptied it out.
    - ◦ else regarding the presently read audio frame as non-proper and then forwarding the presently read audio frame to an output buffer of the memory unit after having performed step (b) on the contents of the speech or pause segment buffer as a segment and emptied it out;
- step (b).1 comprises:
  - identifying peaks of a vowel probability function for the speech segment as candidate vowels; and
  - determining vowels among candidate vowels as a function of the value of said peaks, and the time between consecutive peaks;
  - Identifying intervowel segments as intervals of the speech segment between two successive vowels;
- the average intervowel distance of the speech segment is calculated in step (b).2 either as the average value of durations of the identified intervowel segments, or by using the Kernel Density Estimation algorithm on the durations of the identified intervowel segments;
- said non-linear stretching transfer function is determined is a logarithm function mapping the average intervowel distance to the target intervowel distance;
- determining said non-linear stretching transfer function in step (b).2 comprises fitting the function $T_{out}(T_{in})=A \cdot ln(T_{in})+B$ such that $T_{out}(T_{avg}) = T_{target}$, $T_{out}(T_{min}) = T_{min}$ and $T_{out}(T_{max}) = T_{max}$, wherein T_{min} and T_{max} are given minimum and maximum expected intervowel distance;
- the input media content also comprises an input video signal constituted by a sequence of video frames, the input audio signal and the input video signal being synchronized, step (c) comprising generating an output video signal synchronized with the output audio signal by duplicating video frames when needed;
- step (b).3 comprises for an intervowel segment:
  - determining a target number of samples to be generated for the intervowel segment using the determined stretching transfer function;
  - applying stretching process to update the audio frames of the intervowel segment up to reach said target number of samples to be generated.
- said stretching process comprises for a frame of the intervowel segment:
  - calculating a pitch period of the frame;
  - for each successive pitch period portion of the frame:

◦ appending a copy of the next pitch period portion to said pitch period portion;
◦ stretching the two pitch periods portion by performing a linear cross-fade between the two pitch periods portion and the copy shifted by a pitch period so as to obtain a three pitch periods portion;
◦ extracting the two first pitch periods portion for output, and sending the remaining pitch period back to the input of the stretching process.

- said stretching process when repeated comprises for a frame of the intervowel segment:
  - calculating a pitch period of the frame;
  - for each successive pitch period portion of the frame:
    ◦ appending a copy of the next pitch period portion to said pitch period portion;
    ◦ stretching the two pitch periods portion by performing a linear cross-fade between the two pitch periods portion and the copy shifted by a pitch period so as to obtain a three pitch periods portion;
    ◦ extracting all three pitch periods for output.
- the two pitch periods portion is stretched in a first iteration of the starching process only if the two pitch period portion is a silence, else the two pitch periods portion is stretched in a second iteration of the stretching process only if lag and/or amplitude of an autocorrelation peak of the two pitch periods portion is within a given range;
- step (b).3 comprises for an intervowel segment:
  - determining a target number of samples to be generated for the intervowel segment using the determined stretching transfer function;
  - dividing the intervowel segment into a sequence of periodic, aperiodic or non-stretchable segments, and distributing said target number of samples to be generated between the periodic and aperiodic segments;
  - applying a periodic segment stretching process to update the audio frames of the periodic segments and an aperiodic stretching process to update the audio frames of the aperiodic segments up to reach said target number of samples to be generated.
- said periodic segment stretching process comprises for a periodic segment:
  - calculating a pitch period of the periodic segment;
  - for each successive pitch period portion of the periodic segment:
    ◦ extracting said pitch portion for output;
    ◦ determining a target number of samples to be generated for the pitch period;
    ◦ stretching said pitch period portion by performing a linear cross-fade between said pitch period portion and the next pitch period portion of the periodic segment so as to obtain a two pitch periods portion.
- Said periodic segment stretching process further comprises for each successive pitch period portion of the periodic segment, after stretching said pitch portion:
  ◦ if said target number of samples to be generated for the pitch is exceeded, then further extracting a suitable number of the next samples of the periodic segment for output;
  ◦ else if said target number of samples to be generated for the pitch period is not reached, then for the next pitch period of the periodic segment:
    ▪ extracting a suitable number of samples of said next pitch period portion for output;
    ▪ stretching the last complete pitch period portion in the output by performing a linear cross-fade between said pitch period portion and the next pitch period portion of the periodic segment so as to obtain a two pitch periods portion;
- the pitch period portion is stretched only if the lag and/or amplitude of an autocorrelation peak of the pitch period portion is within a given range;
- said aperiodic segment stretching process comprises for an aperiodic segment:
  - locating zerocrossings of the aperiodic segment;
  - for each successive interval extending between a predetermined number of consecutive zerocrossings of the aperiodic segment:
    ◦ extracting said interval for output;
    ◦ determining a target number of samples to be generated for the interval;
    ◦ checking whether said interval is truly aperiodic;
    ◦ if so, stretching said interval by performing a linear cross-fade between said interval and the next interval extending between said predetermined number of consecutive zerocrossings of the aperiodic segment;
- said aperiodic segment stretching process compris-

es for each successive interval extending between a predetermined number of consecutive zerocrossings of the aperiodic segment, after stretching said interval:

- If said target number of samples to be generated for the interval is exceeded, then further extracting an interval extending between a suitable number of consecutive zerocrossings of the aperiodic segment for output;
- Else if said target number of samples to be generated for the interval is not reached, then:
  - extracting an interval extending between a suitable number of consecutive zerocrossings of the aperiodic segment for output, inferior to said predetermined number;
  - stretching the last interval extending between said predetermined number of consecutive zerocrossings in the output by performing a linear cross-fade between said interval and the next interval extending between said predetermined number of consecutive zerocrossings of the aperiodic segment so as to obtain new samples;

- checking whether an interval is truly aperiodic consists in determining if a length of said interval is lower than a threshold;
- step (b) also comprises, for each pause segment:

  1. calculating an output segment duration T_{out}, based on the segment duration T_{in} and the average intervowel distance T_{avg} and target intervowel distance T_{target} from the previous speech segment;
  2. stretching the pause segment using the determined output duration, so as to generate updated audio frames of the pause segment.

[0018] In a second aspect, the invention provides an equipment comprising a processing unit configured to perform:

- a module for receiving input media content comprising at least an input audio signal constituted by a sequence of audio frames;
- a module for classifying of the audio frames as speech, non-speech, or pause, so as to divide said audio signal of the media content into speech segments bounded by non-speech segments;
- A module for dividing each speech segment into a sequence of intervowel segments;
- A module for calculating an average intervowel distance of each divided speech segment, and for determining a non-linear stretching transfer function mapping an input intervowel distance to an output intervowel distance as a function of said average intervowel distance and a given target intervowel distance;
- A module for stretching each intervowel segment using the determined stretching transfer function so as to generate updated audio frames of the intervowel segment;
- A module for generating an output media content comprising as output signal the input audio signal wherein for each intervowel segment of each speech segment the corresponding audio frames have been replaced by the updated audio frames.

[0019] Preferred but non limiting features of the present invention are as follow:

- The equipment comprises a display for displaying the generated media content.

[0020] According to a third and a fourth aspect, the invention proposes a computer program product, comprising code instructions for executing a method according to the first aspect for slowing down speech in an input media content; and a computer-readable medium, on which is stored a computer program product comprising code instructions for executing a method according to the first aspect for slowing down speech in an input media content.

BRIEF DESCRIPTION OF THE DRAWINGS

[0021] The above and other objects, features and advantages of this invention will be apparent in the following detailed description of an illustrative embodiment thereof, that is to be read in connection with the accompanying drawings wherein:

- Figure 1 represents a domestic installation in which the method according to the invention can be performed;
- Figure 2 represents an architecture of an equipment for performing the method according to the invention;
- Figure 3 is a block diagram illustrating a preferred embodiment of the method according to the invention;
- Figure 4 is a block diagram illustrating speech/non-speech/pause classifying of a preferred embodiment of the method according to the invention;
- Figure 5 is a block diagram illustrating a speech or pause segment processing of a preferred embodiment of the method according to the invention;
- Figure 6 represents an example plot of a vector of calculated segment vowel probabilities with identified peaks and dips, and the peaks detected to represent vowels;
- Figure 7 represents an example of a stretching curve;
- Figures 8a-8b are block diagrams illustrating an intervowel segment processing of first and second pre-

ferred embodiments of the method according to the invention;

- Figures 9a-9b represent respectively first and second iteration of peak detection in an example speech segment's autocorrelation and the choice for pitch period;
- Figure 10 is a block diagram illustrating an audio stretch processing of the first preferred embodiment of the method according to the invention;
- Figure 11 a represents an example audio frame, from which a two pitch period portion is extracted, the autocorrelation function of this portion is calculated, the pitch period peak is found and its position and amplitude is checked if it is within the set range;
- Figure 11b represents an example of crossfading a segment comprising two pitch periods, by applying a fade-out to the second pitch period, then a fade-in to the first pitch period in its shifted copy, and finally summing them together to generate an output segment comprising three pitch periods instead of two;
- Figure 12 is a block diagram illustrating a simplified audio stretch processing in the case of multiple iterations on a single intervowel segment, of the first preferred embodiment of the method according to the invention;
- Figures 13a-13b is a block diagram illustrating an audio stretch processing in the case of periodic / aperiodic intervowel segments, of the second preferred embodiment of the method according to the invention;
- Figures 14a-14b represent examples of excess / insufficient stretching and how it's handled when processing intervowel segments;
- Figure 15 is a block diagram illustrating output processing of a preferred embodiment of the method according to the invention;
- Figure 16 represents an example of the mapping between the original and the output stretched streams.

DETAILED DESCRIPTION OF A PREFERRED EMBODIMENT

[0022] Referring to the drawings, a method according to a possible embodiment of the invention will now be described.

System overview

[0023] The present method aims to slow down speech in an input media content comprising a received input audio signal, i.e. generating an output media content comprising as output signal the input signal which has been processed so to have a speed that is more comfortable for the listener.

[0024] In a preferred embodiment, the output audio signal is slowed down but the intonation and the spectrum are kept as similar as possible to the input audio signal. However, as it will be explained, the audio signal may undergo further treatments so as to change for example the pitch.

[0025] The input media content may only comprise an audio signal, or comprise both an audio and a video signal (visual). Typically, the input media content is a TV stream. In particular, the content format may be MPEG, H.264 or other formats, and may also comprise data that is not compressed.

[0026] As depicted by **figure 1,** the present method is performed by an equipment 1, which may be either an equipment able to directly play the output media content (for example a television, a computer, a smartphone, a tablet, etc.) or an equipment that outputs the output media content to another one which receives it and plays it (for example a set-top box, a server, etc.).

[0027] In the example of figure 1, the equipment 1 is connected to a display 2, and the input media content is streamed from a server 3 connected to the equipment 1 through a network.

[0028] A proposed architecture of the equipment 1 is given by **figure 2.** As presented, the equipment 1 comprises a processing unit 11, such as a processor, and a memory unit 12 for the buffers, such as RAM. The equipment 1 further comprises a user interface 13 for controlling it.

[0029] As already explained, the input media content comprise an audio input signal and advantageously a video audio signal. The input audio signal is constituted of a sequence of audio frames, and similarly the input video signal is constituted of a sequence of video frames (image). Each video frame has an accompanying audio frame. In the represented example a decoder decodes the data from the input data stream and outputs data packets each containing one video frame and the corresponding frame of audio samples. The number of samples in each audio frame depends on the frame rate of the video signal Fr and the sampling rate of the audio Fs, and is equal to their ratio *Fs* /*Fr.* A data packet is output from the decoder every 1/Fr seconds. The decoder does not necessarily decompress the video data, if frame duplication is possible in the coded video stream.

[0030] The Input Buffers block comprises two ring buffers that store the frames of video and audio data. On demand of the processing unit 11, the Input Buffers forward these frames to it.

[0031] The processing unit 11 in turn, advantageously keeps track of how many frames are in the Input Buffers in order to prevent overflow and data loss.

[0032] As it will be shown, the processing unit 11 performs the present method as to modify the audio signal by stretching the parts comprising speech, so that the speech rate is reduced to a target speech rate.

[0033] As it will be shown, the processing unit 11 may produce several output streams that correspond to a set of such target speech rates. For each stream of processed audio, this block advantageously also generates a stream of video that is accordingly modified to maintain synchronization. The audio/video data streams are out-

put in frames to the Output Buffers, and the processing unit 11 makes sure that the output buffers always contain data.

[0034] The Output Buffers store the data frames from the different processed streams and output one to the Coder every *1*/*Fr* seconds, based on the settings made in the User Control interface 13. The interface enables the user to select a desired output speech rate, in syllables per second (syll/s), and optionally an output lowering of the pitch, as well as to rewind the video, and fast forward through it.

[0035] The user can fast forward the processed stream only if there is a time lag between it and the original input stream, introduced by the stretching process. The Encoder on the output end encodes the processed data back into the original data format, applying the appropriate re-compression to the audio signal and the video signal if necessary.

Speech/*Non-speech*/*Pause classification*

[0036] The functional schematic of a preferred embodiment of the processing unit 11 is shown in **figure 3.**

[0037] The present method starts with a step (a) of classifying the audio frames as speech, non-speech, or pause, so as to divide said audio signal of the media content into speech segments bounded by non-speech segments.

[0038] A "silence" is a frame without sound (a silent frame), a "speech" frame is one whose audio is speech, and a "non-speech" frame applies to anything which is not silence or speech, such as music or noise. A silent frame among other silence frames is referred to as a "pause" frame.

[0039] When said input media content is received as a stream, audio frames are being successively read from the input buffer of the memory unit 12 of the equipment 1 to be classified.

[0040] Then, each audio data frame read from the input buffers is preferably stored in an Auxiliary Circular Buffer (implemented within the processing unit 11). This buffer stores a set of audio frames from which the central one is the current frame to be processed (as it will be seen, the other neighbouring audio frames to the current one are preferably needed for Speech/Non-speech/Pause classification). Preferably the entire content of the Auxiliary Circular Buffer is normalized by the Peak Normalization block. This block amplifies the audio signal using an adaptive gain in the range of 0 -12 dB, that updates with each audio frame using a gain step factor seeking to amplify the signal up to the maximum 0 dBFS.

[0041] The block schematic of the classifier of figure 3 is represented with more detail in **figure 4.** As shown, step (a) thus advantageously comprises for each audio frame (from the Auxiliary Circular Buffer):

- determining whether the audio frame is silent by analysing the energy of the audio signal within the frame;
- if the audio frame is not determined as silent, classifying it as speech or non-speech.

[0042] The determination of an audio frame being silent can be made based on the following features:

- frame energy - which is an absolute measure of the energy of the audio signal, and
- frame energy drop - which is a relative difference in energy level between a given frame and the maximum energy level found within the previous non-silence audio segment.

[0043] The speech/non-speech classifier preferably uses a neural network comprising at least three layers: one input layer, one hidden layer and one output layer. The input layer of this network is fed features extracted from the audio signal in the current and neighboring frames, and the output layer generates a probability that the content of the input audio signal is speech. The number of neurons in the input layer equals the number of features used. The number of hidden units is a critical parameter, as more neurons increase the network's complexity and thus its performance, but also decrease its power to generalize on unseen data. There is only one neuron in the output layer.

[0044] There are various features that have been found to work well for speech/non-speech classification that could be incorporated:

- Energy of the filtered input signal averaged over a duration of a defined period of time (for example 1 second), in three bands:
    - ◦ Lowpass filtered band - a bandwidth of 0 - 80 Hz can be chosen, in order to avoid the fundamental frequency of the speech signal, which rarely goes below 80 Hz for males,
    - ◦ Bandpass filtered band - a bandwidth of 80 - 900 Hz can be chosen, in order to capture the energy of the first harmonics and formants in the speech signal, and
    - ◦ Highpass filtered band - a bandwidth of 900 - 20 000 Hz can be chosen, that covers the rest of the signal.
- Modified Low Energy Ratio (MLER) - this gives the ratio of frames with energy lower than a set percentage of the maximum within a defined period of time in the input audio signal or a processed version thereof. For example, a threshold of 15% and a duration of 1s are found suitable. The MLER is calculated on the bandpass filtered input signal.
- Spectral centroid - this gives the "center of mass" of the spectrum, and it is perceptually related to the perceived "brightness" of a sound.

[0045] To train the neural network, a training database of speech and non-speech recordings is used. Training is stopped when the network's ability to generalize degrades for 6 epochs in a series, as determined using a cross-validation set. The trained neural network's performance is then assessed using a test set.

[0046] It would be clear to those versed in the art that different parameters can be used for the presented features, as well as multiple bands. Other features can also be used, e.g. Root Mean Square (RMS), Zero Crossing (ZC) Rate, RMS - ZC Correlation, Spectral Rolloff, Spectral Flux, Mel Frequency Cepstral Coefficients (MFCCs), Voiced to Unvoiced Ratio, Modulation Spectrum, etc. Also, different classification methods can be used, including, but not limited to: Hidden Markov Models (HMM), Gaussian Mixture Models (GMM), and Support Vector Machines (SVM), K Nearest Neighbors (KNN), as well as combinations thereof.

[0047] To refine the final classification decision about the central frame, smoothing is applied to it based on the decisions made for a subset of frames. In other words, step (a) further comprises:

- if the audio frame is determined as silent, classifying it as pause if it is a part of a procession of a plurality of silent audio frames, else classifying it as speech or non-speech according to neighbouring classified audio frames. In particular the Smoothing block finally classifies a central frame determined as silence as a pause if it is a part of a procession of at least 5 frames of silence, corresponding to 200 ms for a 25 fps frame rate Fr.

[0048] One simple approach to smoothing is to make a decision for the current frame based on an average of the decisions made for neighboring frames. Another is to take into account the decisions for the neighboring frames using weights that are a function of their distance from the current frame.

[0049] The schematic includes a data gate and a demultiplexer block that are introduced to control the flow of audio/video frames according to the outputs of the decision blocks.

[0050] Based on the classification decision and as represented by figure 3, the current frame is preferably evaluated as proper or not, so as to control the gate.

[0051] Thus step (a) further comprises:

- if the presently read audio frame is classified as speech or pause and if the previously read audio frame is identically classified, regarding it as proper and forwarding it (using the demultiplexer) to a speech or pause segment buffer of the processing unit 11;
- else if the presently read audio frame is classified as speech or pause and if the previously read audio frame is not identically classified (i.e. if the current frame is speech and the previous one was a pause, or vice versa), regarding it as not proper and forwarding it to an internal speech or pause segment buffer in the processing unit 11, after having performed next step (b) of processing the contents of the speech or pause segment buffer as a segment and having emptied it out. In other words the data gate is temporarily closed and the contents of the Speech or Pause Segment Buffer are processed and output, after which the data gate is reopened and the current frame is forwarded by the demultiplexer to the Speech or Pause Segment Buffer.
- else regarding the presently read audio frame as non-proper and then forwarding the presently read audio frame to an output buffer of the memory unit 12 after having performed step (b) on the contents of the speech or pause segment buffer as a segment and emptied it out. In other words the data gate is temporarily closed and any contents of the Speech or Pause Segment Buffer is forwarded to the process Speech or Pause Segment block, which in turn outputs it to the write Output Buffers block. Only then is the data gate reopened and the current non-proper frame is forwarded by the demultiplexer to the write Output Buffers block.

[0052] The Speech or Pause Segment Buffer can contain either speech or pause, as its name implies. The process Speech or Pause Segment block processes these frames differently according to their type, as it will be now explained.

Speech or pause processing

[0053] In a main step (b), the method comprises, for each speech segment:

1. dividing the speech segment into a sequence of intervowel segments;
2. calculating an average intervowel distance T_{avg} of the speech segment, and determining a non-linear stretching transfer function mapping an input intervowel distance T_{in} to an output intervowel distance T_{out} as a function of said average intervowel distance T_{avg} and a given target intervowel distance T_{target};
3. for each intervowel segment of the speech segment, stretching the intervowel segment using the determined stretching transfer function so as to generate updated audio frames of the intervowel segment.

[0054] Advantageously, in the main step (b), the method also comprises, for each pause segment:

1. calculating an output segment duration T_{out}, based on the segment duration T_{in} and the average intervowel distance T_{avg} and target intervowel distance T_{target} from the previous speech segment;
2. stretching the pause segment using the deter-

mined output duration, so as to generate updated audio frames of the pause segment.

[0055] The internal structure of the Process Speech or Pause Segment block performing said step (b) is shown in **figure 5.**

[0056] An "intervowel segment", is a fragment of a speech segment between two successive vowels. In the case of a pause segment, the whole segment is treated as a single intervowel segment.

[0057] If the Speech or Pause Segment Buffer contains speech it is first analyzed by the Calculate Syllable Rate block as to perform step (b).1.

[0058] The Calculate syllable rate block estimates the average intervowel segment length, which corresponds to the average syllable rate of the speech segment, by first locating the positions of the vowels in it and then calculating the intervowel distances between them.

[0059] More precisely, step (b).1 comprises:

- identifying peaks in the vowel probability function of the speech segment as candidate vowels; and
- determining vowels among candidate vowels as a function of the absolute and relative values of the peaks and the time between consecutive peaks;
- Identifying intervowel segments as intervals of the speech segment between two successive vowels.

[0060] Indeed, a vowel is a sound in spoken language, pronounced with an open vocal tract, so that the tongue does not touch the lips, teeth, or roof of the mouth. This contrasts with consonants, which have a constriction or closure at some point along the vocal tract. Furthermore, a vowel carries the peak energy in a syllable.

[0061] Vowels have therefore acoustical characteristics that separate them from consonants, including pronounced periodicity and a concentration of energy in the lower frequency bands. In order to identify them a neural network can be used that uses a set of Mel Frequency Cepstral Coefficients (MFCCs) extracted from short segments (frames) of the signal as input, and outputs the probability of that frame being a vowel. Next, local peaks and dips are identified in the vector of the output vowel probabilities. The identified peaks are declared to be vowel candidates. An example plot of the vowel probability function and the identified peaks and dips is shown in **figure 6.**

[0062] In a preferred embodiment, from the identified vowel candidates only the peaks that meet the following conditions are declared to represent vowels:

1. The absolute value of the peak is greater than a predetermined value.
2. The difference between the peak and its preceding dip is greater than a predetermined value.
3. A sufficient amount of time has passed from the last peak that was declared to be a vowel.

[0063] Peaks declared to represent vowels are presented with vertical lines in figure 6.

[0064] It is clear to one versed in the art that any other machine learning algorithm can be employed to the aim of finding the vowel positions. Moreover, other features of the signal can be used to the same end, whether on their own or in combination with each other. Such features include but are not limited to the amplitude envelope, the Low Frequency Modulated Energy (LMFE) and the high-to-low frequency energy ratio (ER).

[0065] Once all of the intervowel distances have been determined, they are analyzed to estimate the average intervowel distance T_{avg} of the speech segment, and possibly the average syllable rate for the current speech segment: $SR_{avg} = 1/T_{avg}$.

[0066] This can be done by simply calculating T_{avg} as the average of all of the intervowel distances T_{iv}. Another approach is to use the Kernel Density Estimation (KDE) algorithm to estimate T_{avg} using Gaussian kernels fitted to the histogram of the intervowel distances T_{iv}. The largest peak is then chosen to be the average intervowel distance T_{avg}.

[0067] The average intervowel distance T_{avg}(or the average syllable rate $SR_{avg} = 1/T_{avg}$) for the current speech segment is used to generate the stretching curve, defining said non-linear stretching that maps an input intervowel distance T_{in} to an output intervowel distance T_{out} as a function of said average intervowel distance T_{avg} and a given target average intervowel distance T_{target}. The stretching curve will be used to process the audio data in order to obtain an audio stream corresponding to the target speech rate.

[0068] In the current implementation in addition to the original stream a plurality of reduced speech rate streams are generated at different target speech rates, in particular at 3, 4 and 5 syll/s, or in other words a T_{target} of 1/3, 1/4, 1/5 s. In addition, each output stream may be processed with a resampling based pitch shift to generate a plurality of streams with different levels of pitch lowering, in particular 3. In the present example, this gives a total of (3 syllable rates + 1 unaltered) x (3 pitch lowerings + 1 unaltered) = 16 output streams.

[0069] The Calculate stretching curve block uses the average intervowel distance T_{avg} to calculate the stretching transfer function of which an example is shown in **figure 7.**

[0070] This function defines the mapping between the intervowel distances in the input speech segment T_{in} and the output intervowel distance targets T_{out} that need to be obtained using stretching in order to reach the set target intervowel distance T_{target}.

[0071] In the example of figure 7, the curve is generated for an input average intervowel distance T_{avg} of 166 ms and a target syllable rate of 3 syll/s, corresponding to a target average intervowel distance T_{target} of 333 ms. Also shown are a line for no stretching ($T_{out} = T_{in}$, green dotted line) and a line for double stretching ($T_{out} = 2T_{in}$, black dotted line).

[0072] The nonlinearity of the stretching curve assures larger stretching of smaller intervowel distances, which are harder to perceptually process, and less stretching for longer ones, which do not impede comprehension. Additionally, this nonlinearity deals effectively with bursts of increased speech rate that might be embedded within the speech segment.

[0073] Said non-linear stretching transfer function in the preferred implementation is determined as a logarithm function mapping the average intervowel distance T_{avg} to the target intervowel distance T_{target}.

[0074] Preferably, determining said non-linear stretching transfer function in step (b).2 comprises fitting the function $T_{out}(T_{in}) = A \cdot ln(T_{in}) + B$ such that $T_{out}(T_{avg}) = T_{target}$, $T_{out}(T_{min}) = T_{min}$ and $T_{out}(T_{max}) = T_{max}$, wherein T_{min} and T_{max} are given minimum and maximum expected intervowel distance.

[0075] More precisely, the stretching transfer function can be calculated by the least squares fitting of a logarithm function:

$$T_{out}(T_{in}) = A \cdot ln(T_{in}) + B$$

to the three points (T_1, T_{out1}), (T_2, T_{out2}) and (T_3, T_{out3}) defined as:

$$T_1 = T_{out1} = T_{min}$$

$$T_2 = T_{avg}$$

$$T_{out2} = T_{target}$$

$$T_3 = T_{out3} = T_{max}$$

[0076] At the end, the fitted logarithm function is advantageously compared with a line function with a slope k equal to 1 (i.e. the identity function), shown in figure 7, so that the final stretching curve is a maximum of the two functions, i.e. $T_{out}(T_{in}) = \max(A \cdot ln(T_{in}) + B, T_{in})$. This is done to avoid requiring shortening of the speech audio.

[0077] In order to assure maximum output quality, it is preferred to not let the stretching factor go above the maximum stretching factor of 2. To do this, the stretching curve is advantageously further compared with a line with slope k equal to 2 which goes through the origin, also shown in figure 7, so that the final stretching curve is the minimum of the two functions, i.e. $T_{out}(T_{in}) = \min(A \cdot ln(T_{in}) + B, 2 \cdot T_{in})$.

[0078] An additional possible enhancement of the stretching curve is its adaptation to current speech dynamics through temporal group analysis of consecutive detected intervowel distances. This helps further targeting bursts of fast speech embedded in periods of slow speech. It also potentially discovers insertion errors made by the vowel detection algorithm.

[0079] Once the stretching curve has been calculated the speech segment is split in intervowel segments based on the determined vowel locations, and it is processed one intervowel segment at a time in step (b).3.

[0080] These intervowel segments are extracted by the Extract Intervowel Segment block that writes them to the Intervowel Segment Buffer and are stretched by the Process Intervowel Segment block, which sends its output to the Write to Output Buffers block. This procedure is repeated until the whole speech segment is processed.

Intervowel segment processing - First embodiment

[0081] The structure of a first embodiment of a Process Intervowel Segment block performing said step (b).3 is shown in **Fig. 8a.**

[0082] In this embodiment, for an intervowel segment to be stretched:

- a target number of samples to be generated for the intervowel segment is determined using the determined stretching transfer function;
- a stretching process is iteratively applied to update the audio frames of the intervowel segment up to reach said target number of samples to be generated.

[0083] To this end a target duration of the segment is calculated by the stretching curve, from which the number of audio samples to be generated for the whole segment is calculated. Frames of the audio segment are stored for processing in a Work Buffer by the Create Work Buffer block.

[0084] Then, a target number of samples to be generated for the current audio frame in the Work Buffer is calculated by the Update Number of Frame Samples to Generate block. This block updates the target number of samples to generate for the current frame, with the number of samples that were targeted but not achieved in the stretching of the previous frame. This number is input into the Stretch Audio block together with a calculated pitch period (see below).

[0085] Indeed, said stretching process advantageously comprises for a frame:

- calculating a pitch period of the frame;
- for each successive pitch period portion of the frame:
    - ◦ appending a copy of the next pitch period portion to said pitch period portion;
    - ◦ stretching the two pitch periods portion by performing a linear cross-fade between the two pitch periods portion and a copy shifted by a pitch period so as to obtain a three pitch periods portion.

[0086] To determine the pitch period, pitch extraction approaches are known by the skilled person. Alternatively, the pitch period is determined by performing an iterative procedure as illustrated by the example of **figures 9a-9b.**

[0087] In this iterative procedure, first, an autocorrelation of the speech signal in the frame of audio is calculated. Second, local peaks are located in the autocorrelation function by finding points that are greater than their immediate predecessors and successors. Boundary points cannot be peaks as they have no predecessor or successor. To determine which of the detected peaks corresponds to the pitch period, they are put in a vector, and new peaks are detected in this vector in an iterative procedure that ends when there are less than 3 peaks left. In the example, figure 9a shows the first iteration and figure 9b the second iteration of peak detection.

[0088] From the peaks output at the last iteration the highest one is selected. The pitch period is then calculated as the distance between the central peak of the autocorrelation and the lag of said selected peak.

[0089] Advantageously, if the calculated pitch period is outside set minimum and maximum pitch periods (defining expected bounds of the pitch in speech), the result is discarded and the previously determined pitch period is used. The bounds may be set to 3 and 20 ms, which correspond to a pitch range between 50 and 333 Hz.

[0090] Then, the Stretch Audio block stretches the current frame. The Stretch Audio block's internal structure is shown in **figure 10.**

[0091] As already mentioned, the audio frame stored in the Work Buffer is preferably analyzed pitch period by pitch period, i.e. one pitch period long portions of the frame are considered. At the start, the beginning pitch period portion is transferred to the First Pitch Period Buffer. Then the next one is appended to it, and both are transferred to the Two Pitch Periods Buffer. An example content of the Work Buffer and the Two Pitch Periods Buffer is shown in **figure 11 a.**

[0092] These two pitch periods can then be processed to generate three pitch periods if they satisfy the necessary conditions.

[0093] This block preferably stretches the audio frame in the Work Buffer in a two iteration process that assures maximum stretching quality. In the first iteration the audio frames containing silence are stretched. Stretching silence periods gives almost no noticeable processing artifacts. In the second iteration frames with pronounced periodicity are stretched, as they are favorable for processing with the Pitch Synchronized Overlap Add (PSOLA) based algorithm.

[0094] To this end, first the two pitch period portion is identified as silence or speech. Then its autocorrelation is calculated, also shown in figure 11 a. If it is speech, the lag of the maximum autocorrelation peak above the minimum pitch period lag is determined. If it is silence, then the maximum autocorrelation peak above the current pitch lag is determined. This information is used to evaluate the suitability of the data for stretching. The data will be advantageously stretched only if the target number of samples to be generated has not been reached and either of the following two conditions are satisfied:

1. If it is the first iteration of the Stretch Audio block and the two pitch period portion is a silence.
2. If it is the second iteration of the Stretch Audio block and the data is speech, and additionally the lag and amplitude of the autocorrelation peak are within the set range, as illustrated with the red rectangle in figure 11a. This assures pronounced periodicity in the speech signal.

[0095] Both conditions assure for minimal processing artifacts in the output speech. Moreover, since the duration of pauses and vowels changes noticeably in naturally slow speech, these conditions contribute to the naturalness of the output speech.

[0096] If the data is to be processed, the Linear Cross Fade block stretches as explained the audio signal by generating a linear cross-fade between the two pitch period portion and its copy shifted by the calculated overlap, as illustrated in **figure 11 b.** This process results in a segment of three pitch periods instead of two.

[0097] After the stretching, in a first case the first two pitch periods are split from the third one and output to the Processed Work Buffer. The third leftover pitch period is transferred back to the First Pitch Period Buffer where a new pitch period will be appended to it. When the two pitch period portion is not stretched, the two pitch periods are also split, the first one is output through the generate output block, and the second one is again returned to the First Pitch Period Buffer. If not all pitch periods from the Work Buffer have been processed, the algorithm repeats.

[0098] In a second case, the Stretch Audio block is run an additional time by the Process Intervowel Segment block, which is the case when the target number of additional audio samples to be generated was not reached, the Stretch Audio block's internal structure simplifies to the one shown in **figure 12.** All three pitch periods are extracted for output. This assures that successions of artificially generated pitch periods are kept at minimum length. Namely, if the processed intervowel segment is input to the Stretch Audio block in the Process Intervowel Segment block a second time, then a maximum of two generated consecutive pitch periods can occur in its output. If it is passed an additional third time, which is rarely needed, then this maximum grows to three consecutive pitch periods.

[0099] In case of an audio/video signal, the processed output (i.e. the stretched audio frame) is sent to the Generate Output Audio/Video Frame block. This block accumulates the processed audio data, uses it to construct audio frames and combines these audio frames with the corresponding video frame received from the Create Work Buffer block. If there is more than one audio frame of accumulated data, multiple audio frames are created

and combined with copies of the same video frame. In this way synchronization between the audio and video streams is maintained.

[0100] This process repeats until the whole intervowel segment is processed. At the end if the target number of samples to generate is not reached the output processed intervowel segment stored in the Processed Intervowel Segment Buffer is treated as input and the whole process repeats. The Update Intervowel Number of Samples to Generate block calculates the new target number of samples to generate. If the contents of the Processed Intervowel Segment Buffer are the same with that of the input Intervowel Segment Buffer at the end of the processing, and additional samples need to be generated, the stretching criteria for pronounced periodicity evaluated in the Stretch Audio block are relaxed.

Intervowel segment processing - second embodiment

[0101] The structure of a Process Intervowel Segment block performing a second embodiment of said step (b).3 is shown in **Fig. 8b.**

[0102] In this embodiment step (b).3 comprises for an intervowel segment:

- determining a target number of samples to be generated for the intervowel segment using the determined stretching transfer function;
- dividing the intervowel segment into a sequence of periodic, aperiodic or non-stretchable segments, and distributing said target number of samples to be generated between the periodic and aperiodic segments;
- applying a periodic segment stretching process to update the audio frames of the periodic segments and an aperiodic stretching process to update the audio frames of the aperiodic segments to reach said target number of samples to be generated.

[0103] In other words, as in the first embodiment, the current intervowel segment is also stretched in an iterative manner to the target duration as calculated by the stretching curve.

[0104] However, in this embodiment, the segment is evaluated for stretching, i.e. divided into "elementary segments" (sub-parts of the intervowel segment) such that all of the elementary segments that are favorable for stretching are identified. This can be done using the amplitude and estimated periodicity of the speech signal in each frame of the intervowel segment. Frames are extracted using a sliding window. Advantageously, the elementary segments are specifically classified between those which:

1. Have pronounced periodicity. These elementary segments, referred to as "periodic segments" are voiced sounds, such as vowels, and can be stretched with high quality using an algorithm for periodic stretching, in particular a PSOLA based algorithm.
2. Have low or no periodicity. These elementary segments, referred to as "aperiodic segments", are often fricatives and affricate consonants, and can be stretched with high quality using an algorithm for aperiodic stretching. Plosives are naturally excluded from this class, by way of using a longer sliding window in the analysis.

[0105] Elementary segments which have low amplitude, i.e. represent silence are treated as aperiodic segments. These elementary segments can be stretched without generating audible artifacts.

[0106] Other elementary segments are considered as "non-stretchable segments".

[0107] For assessing the periodicity, also known as probability of voicing, there are a range of approaches known in the prior art, including methods based on the autocorrelation, spectrum of the signal, signal models etc. One state-of-the-art method presented in the document Ghahremani, Pegah, Bagher Baba Ali, Daniel Povey, Korbinian Riedhammer, Jan Trmal, and Sanjeev Khudanpur. "A pitch extraction algorithm tuned for automatic speech recognition." In 2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 2494-2498. IEEE, 2014, uses the Normalized Cross Correlation Function (NCCF).

[0108] The portions favorable for stretching are found by thresholding the amplitude and estimated periodicity for each frame. In the preferred embodiment, two thresholds are used for the normalized amplitude of the autocorrelation peak located in the pitch region, in a way in which a high amplitude pitch related peak is used to identify periodic segments favorable for stretching, and a low amplitude peak is used to detect aperiodic segments favorable for stretching.

[0109] The decision is at the end smoothed. In addition to recognizing the segments favorable for stretching, this block also calculates the pitch period for each frame of the signal, in particular by using the method described for the first embodiment (see figures 9a-9b).

[0110] Once these stretchable elementary segments are identified, the number of audio samples to be generated for the whole intervowel segment is distributed equally among them, calculating the number of samples to be generated per input sample for these segments. Next, the algorithm goes through the contents of the intervowel segment buffer that was segmented into stretchable and not stretchable parts. It loads these elementary segments one by one in the Extracted Segment Buffer and processes them accordingly. If the elementary segment has pronounced periodicity (periodic segment) it is forwarded to the process periodic segment block, if it has pronounced aperiodicity (aperiodic segment, including silences as explained), then it is forwarded to the process aperiodic segment block. Finally, if the elementary segment is not favorable for stretching (non-stretchable segment) it is forwarded directly to the Processed Intervowel

Segment Buffer, where the data from the process periodic segment and process aperiodic segment is also output.

[0111] The Process Periodic Segment block's internal structure is shown in **figure 13a.** The periodic segment processing is quite similar to the audio stretching of the first embodiment.

[0112] In other words, said periodic segment stretching process advantageously comprises for a periodic segment:

- calculating a pitch period of the periodic segment (as already explained it can be performed earlier while assessing the periodicity of the elementary segment);
- for each successive pitch period portion of the periodic segment:
    - ◦ extracting said pitch portion for output;
    - ◦ determining a target number of samples to be generated for the pitch portion (based on the target number of samples to be generated per input sample);
    - ◦ stretching said pitch period portion by performing a linear cross-fade between said pitch period portion and the next pitch period portion of the periodic segment so as to obtain a two pitch periods portion.

[0113] More precisely, the first pitch period from the periodic audio segment stored in the Extracted Segment Buffer is output to the Processed Intervowel Buffer, based on the pitch found for that elementary segment. Next, the target number of samples to generate for this pitch period is calculated based on the target number of samples to be generated per input sample. Next, the last pitch period from the Processed Intervowel Segment Buffer is taken and the next pitch period from the Extracted Segment Buffer is appended to it. Both pitch periods are processed by the Generate New Pitch Period Block.

[0114] These two pitch periods can then be processed to generate three pitch periods if they satisfy the necessary conditions. As already explained for the first embodiment, the data will be advantageously stretched only if the target number of samples to be generated has not been reached, and if the lag and amplitude of the autocorrelation peak are within the set range, as illustrated with the red rectangle in figure 11 a. This assures pronounced periodicity in the speech signal.

[0115] If the conditions for stretching are met then the Generate New Pitch Period Block stretches the audio signal by generating a linear cross-fade between the two pitch period portion and its copy shifted by the calculated overlap, as illustrated in figures 11 b. This process results in a segment of three pitch periods instead of two.

[0116] After the stretching, the newly generated pitch period is forwarded to the Processed Intervowel Segment Buffer where it is concatenated to the first pitch period, which was forwarded there previously. When the two pitch period portion is not stretched, nothing is forwarded to the Processed Intervowel Segment Buffer.

[0117] Next, the target number of samples to generate is updated with the samples generated, i.e. if a new pitch period was generated then its length is subtracted from this target.

[0118] If the result of this update is negative, i.e. excessive new samples have been generated, then a suitable amount of input samples are extracted and forwarded directly to the Processed Intervowel Segment Buffer. By "suitable", it is meant in this case the number of input samples that would require a number of target samples to be generated equal to the ones already generated in excess. Such direct forwarding to compensate for the "over-generation" of samples inserts a shift in the pitch period portions, which adds a randomization factor to the stretching that effectively increases the output quality.

[0119] On the other hand, if the result of the update is positive, then we haven't reached the target number of samples to be generated with the new pitch period, and nothing more is done, at this iteration.

[0120] More precisely, and as it will be explained, if said target number of samples to be generated for the pitch period is not reached, then for the next pitch period of the periodic segment, the periodic segment processing comprises:

- extracting a suitable (see below) number of samples of said next pitch period portion for output;
- stretching the last complete pitch period portion of the output (i.e. the complete pitch period of the last extracted samples) by performing a linear cross-fade between said pitch period portion and the next pitch period portion of the periodic segment so as to obtain a two pitch periods portion.

[0121] Next, in the case of audio/video signal, the number of samples accumulated in the Processed Intervowel Segment Buffer is evaluated, and if it has reached a length that corresponds to the video frame rate then a video frame is taken from the Extracted Segment Buffer corresponding to the second pitch period from the two used in the Generate New Pitch Period Block, and it is added to audio stream in the Processed Intervowel Segment Buffer.

[0122] After the synchronization of video frames has been taken care of, the next pitch period is extracted from the Extracted Segment Buffer to the Processed Intervowel Segment Buffer. In the case when not enough new samples have been generated, only a suitable number of samples of the next pitch period are extracted. In this case, the "suitable" number is preferably obtained when the pitch period is reduced by the number of samples that still have to be generated. This allows the algorithm to effectively catch up on the target number of samples to generate the next time it generates a new pitch period. The target number of samples to generate is then updat-

ed for the added input samples to the Processed Intervowel Segment Block. If the end of the Extracted Segment Buffer has been reached the Process Periodic Segment algorithm terminates.

[0123] Two examples cover the cases when there is an excess of generated samples, and when there is an insufficiency of generated samples, in order to clarify the algorithm.

[0124] Example 1 - Excess stretching, **figure 14a.** Assuming that 0.8 new samples have to be generated per input sample and the pitch period is 200 samples, then after extracting the first pitch period to the Processed Intervowel Segment Buffer, 200 * 0.8 = 160 new samples have to be generated for it (1°). If a new pitch period is generated, then its length of 200 samples means that after updating the target number of samples to generate 160 - 200 = -40 samples are obtained, meaning that 40 samples have been generated in excess (1°). To compensate for this 40 / 0.8 = 50 input samples are extracted from the Extracted Segment Buffer directly to the Processed Intervowel Segment Buffer (2°). For these 50 samples the processing means 11 would have had to generate new 50 * 0.8 = 40 samples, but this has already been done, making the number of samples to generate equal to 0. After this, the algorithm extracts the next pitch period from the Extracted Segment Buffer to the Processed Intervowel Segment Buffer (3°). The target number of samples to generate is again updated and it will again equal 200 * 0.8 = 160, assuming the size of the pitch period hasn't changed. The process continues from here in the same fashion. We can see how step (2°) inserts a shift in the pitch period portions which adds a randomization factor to the stretching that effectively increases the output quality.

[0125] Example 2 - Insufficient stretching, **figure 14b.** This time assuming 1.25 new samples have to be generated per input sample and the pitch period is again 200 samples, for the first pitch period in the Processed Intervowel Segment Buffer 200 * 1.25 = 350 new samples have to be generated (1°). A generated new pitch period will have 200 samples, which means that after updating the target number of samples to generate 350 - 200 = 150 additional samples should still be generated (1°). Because of this, when taking the next pitch period from the Extracted Segment Buffer, the processing means 11 shortens it by 150 samples, i.e. extracts only the first 50 samples from it (2°). The target number of samples to generate will now equal 150 + 50 * 1.25 ≈ 212. In order to generate the next new pitch period, a whole pitch period from the Processed Intervowel Segment Buffer is now taken, which will comprise the 50 samples extracted and forwarded from the Extracted Segment Buffer preceded by 150 samples from the pitch period previously generated. The next pitch period from the Extracted Segment Buffer is then taken and a new pitch period is generated and forwarded to the Processed Intervowel Segment Buffer. The update of the target number of samples to generate will now give 212 - 200 = 12 samples that still should be generated. Which means that the next pitch period that will be forwarded to the output will be reduced by 12 samples. And the process continues. Again, we can see how step (2°) introduces a shift in the pitch period portions adding a randomization factor to the stretching.

[0126] The Process Aperiodic Segment Block, see **figure 13b,** works in a similar fashion to the Process Periodic Segment Block, in that instead of working with pitch periods it works with intervals of the speech signal extending between a predetermined number N+1 (for example chosen between ten and fifty, advantageously around thirty) of zerocrossings with a positive slope, referred to as 'N-interzerocrossings interval'. More precisely, each said N-interzerocrossings interval is to be understood as the union of N consecutive sub-intervals each extending between two consecutive zerocrossings, i.e. the union of N '1-interzerocrossings intervals', the latter being referred to simply as an 'interzerocrossing' interval.

[0127] In other words, said aperiodic segment stretching process advantageously comprises for an aperiodic segment:

- locating zerocrossings of the aperiodic segment with positive slope;
- for each successive interval extending between said predetermined number of N+1 consecutive zerocrossings of the aperiodic segment (for instance if a given N-interzerocrossings interval extends between zerocrossings numbers i and i+N, the next N-interzerocrossings interval extends between zerocrossings numbers i+N and i+2N, etc.):
    - ◦ extracting said interval for output;
    - ◦ determining a target number of samples to be generated for the interval;
    - ◦ checking whether said interval is truly aperiodic (Practically, the checking of the aperiodicity may be for two N-interzerocrossings segments, i.e. the one in the output and the next one, see below);
    - ◦ if so, stretching said interval by applying a linear cross-fade to it and the said next such interval of the aperiodic segment (the interval between the next N consecutive zerocrossings), thus generating new samples (M new samples in the schematic of figure 13b).

[0128] More precisely, firstly the speech signal in the Extracted Segment Buffer is analyzed so that all of the zerocrossings with a positive slope are located in it. The algorithm is then initialized by copying the first N-interzerocrossings interval, and any preceding samples, into the Processed Intervowel Segment Buffer. Next, the target number of samples is calculated based on the length of this interval and the target number of samples to generate per input sample.

[0129] The next N-interzerocrossings interval is then

extracted from the Extracted Segment Buffer and forwarded to the Generate new M samples Block, which checks whether the interval is truly aperiodic, by checking its length, in particular by verifying that the length of the signal is below a threshold, which is preferably an upper threshold. Indeed, in addition a lower threshold is advantageously also set in order to: 1) guarantee high-quality in the aperiodic stretching process, which gives worse results with shorter segments, and 2) in the case when the aperiodic segment for stretching represents a short plosive segment this stops it from being processed. It is to be noted that alternatively, said next interval taken from the Extracted Segment Buffer is appended to the last interval forwarded to the Processed Intervowel Segment Buffer, so that the Generate new M samples Block, checks whether the whole resulting interval is truly aperiodic (in other words, as explained the checking of the aperiodicity may be for two N-interzerocrossings segments).

[0130] If it's too large as determined by said upper threshold, this would point to the presence of low frequency content in the interval, and linear cross fading is not applied, i.e. the interval is directly forwarded to the Processed Intervowel Segment Buffer.

[0131] If it is short enough as determined by the (upper) threshold, linear cross fading is applied between it and a shifted copy of itself and the resulting new samples are forwarded to the Processed Intervowel Segment Buffer.

[0132] After this, the target number of samples to generate is updated in a similar fashion to previously in the Process Periodic Segment Buffer. Again, if there is an excess of samples generated, a suitable amount of data is extracted from the Extracted Segment Buffer to the Processed Intervowel Segment Buffer. The difference here is that only whole interzerocrossings intervals are copied in this process. This means that the exact number of samples will almost never be copied, so as few interzerocrossings intervals as needed (i.e. a k-interzerocrossings interval with $k \in \mathbb{N}$ as small as possible) are copied to get a positive target number of samples for generation. As in the periodic stretching, such direct forwarding to compensate for the "over-generation" of samples inserts a shift in the N-interzerocrossings intervals (if k≠N), which adds a randomization factor to the aperiodic stretching that effectively increases the output quality.

[0133] On the other hand, if the target number of samples has not been reached, only a reduced number of interzerocrossings, less than the predetermined number N (i.e. a shorter k-interzerocrossings interval, with k<N) of the aperiodic segment will be extracted and forwarded to the output. The next stretching will be done on the last complete N-interzerocrossings interval in the output Process Periodic Segment Buffer, i.e. the interval constituted of said k-interzerocrossings interval preceded by the sub-interval (a '(N-k)-interzerocrossings interval') of the last whole N-interzerocrossings interval extracted.

[0134] To sum up, the aperiodic segment processing preferably comprises, for an interval extending between N consecutive zerocrossings with positive slope assessed as truly aperiodic:

- stretching said interval by performing a linear cross-fade between said interval and the next such interval of the aperiodic segment so as to obtain new samples, and
- If said target number of samples to be generated for the interval is exceeded, then:
    - ◦ further extracting an interval extending between a suitable number of consecutive zerocrossings of the aperiodic segment for output, else:
        - ◦ extracting an interval between a reduced number of the next consecutive zerocrossings of the aperiodic segment for output, and then stretching the last complete N-interzerocrossings interval from the output (by linear cross-fading between said interval and the next such interval of the aperiodic segment).

[0135] Next, the number of added samples in the Processed Intervowel Segment Buffer is evaluated for adding a video frame from the Extracted Segment Buffer. If it is sufficient, the video frame that corresponds to the last interzerocrossing interval is added.

[0136] The next N-interzerocrossings intervals then extracted and forwarded to the Processed Intervowel Segment Buffer, and the target number of samples to generate is updated accordingly. If the algorithm has reached the end of the Extracted Segment Buffer then it terminates, otherwise the process repeats.

[0137] After each elementary segment is processed the number of generated samples is evaluated in terms of the target number of samples for that segment.

[0138] Finally, after all of the elementary segments have been processed, the number of generated intervowel samples is checked. If it has been reached, then the contents of the Processed Intervowel Segment Buffer are forwarded to the Write to Output Buffers Block. If not, then the whole process is repeated with the contents of the Processed Intervowel Segment Buffer forwarded to replace the contents of the Intervowel Segment Buffer. The number of samples to be generated at this iteration is updated accordingly. If the target number of samples has not been reached in two iterations of stretching, than the selection criteria used in the recognition of segments favorable for stretching, and the criteria used to check for periodicity and aperiodicity in the stretching block are relaxed. This produces more and larger segments for stretching, albeit reducing the quality of the stretching.

Outputting the processed data

[0139] In a final step (c) the processing unit 11 generates an output media content comprising as output signal the input audio signal wherein for each intervowel segment of each speech segment the corresponding audio frames have been replaced by the updated audio frames.

[0140] In the case of video signal, step (c) also comprises generating an output video signal synchronized with the output audio signal by duplicating video frames when needed, as explained.

[0141] To this end, the Read from Output Buffers block controls what is sent to the encoder that generates the output data stream. The internal structure of the Read from Output Buffers block is shown in **figure 15.**

[0142] This block advantageously further gives the user three functionalities, which can be accessed through the user interface 13 (the User Control block):

1 To set the desired output speech rate through the syll/s parameter. This user input determines which from the several processed audio/video streams will be read from the Output Buffers and thus output from the system. In order to facilitate seamless switching between the different streams, first the original audio/video frame is found that corresponds to the current frame in the present stream, and then the output is redirected to its position in the requested stream. An example of the mapping between the original and the output stretched streams is shown in **figure 16.**
2 Activate an optional pitch change in the output audio stream. Since this is done by the resampling of the audio/video stream, the process shifts the whole spectrum of the audio signal towards the low frequencies. This can potentially improve intelligibility for people with high frequency hearing loss, i.e. presbycusis, as found in the elderly.
3 Rewind (RW) or Fast-Forward (FF) through the processed audio/video data. This is easily accomplished through changing the location to be read from the Output Buffers via updating the memory pointer. FF is only possible if there is a built up delay between the currently output frame of the slowed-down data stream and the frame currently output by the Processing Unit in that stream. These two options will not process the audio in any way and will only skip through it.

Claims

1. A method for slowing down speech in an input media content received by an equipment (1) comprising a processing unit (11), the input media content comprising an input audio signal constituted by a sequence of audio frames, the method being **characterized in that** it comprises performing by the processing unit (11) steps of:

   (a) classifying the audio frames as speech, non-speech, or pause, so as to divide said audio signal of the media content into speech segments bounded by non-speech segments;
   (b) for each speech segment:

      1. dividing the speech segment into a sequence of intervowel segments;
      2. calculating an average intervowel distance (T_{avg}) of the speech segment, and determining a non-linear stretching transfer function mapping an input intervowel distance (T_{in}) to an output intervowel distance (T_{out}) as a function of said average intervowel distance (T_{avg}) and a given target intervowel distance (T_{target});
      3. for each intervowel segment of the speech segment, stretching the intervowel segment using the determined stretching transfer function so as to generate updated audio frames of the intervowel segment;

   (c) generating an output media content comprising as output signal the input audio signal wherein for each intervowel segment of each speech segment the corresponding audio frames have been replaced by the updated audio frames.

2. A method according to claim 1, wherein step (a) comprises for each audio frame:

   - determining whether the audio frame is silent by analysing the energy of the audio signal within the frame;
   - if the audio frame is not determined as silent, classifying it as speech or non-speech.

3. A method according to claim 2, wherein step (a) further comprises for each audio frame:

   - if the audio frame is determined as silent, classifying it as pause if it is a part of a procession of a plurality of silent audio frames, else classifying it as speech or non-speech according to neighbouring classified audio frames.

4. A method according to any one of claims 1 to 3, wherein said input media content is received as a stream, audio frames being successively read from an input buffer of a memory unit (12) of the equipment (1) to be classified, wherein:

   - Step (b) further comprises forwarding the generated updated audio frames to an output buffer;
   - step (a) further comprises:

      ◦ if the presently read audio frame is classified as speech or pause and if the previously

read audio frame is identically classified, regarding it as proper and forwarding it to a speech or pause segment buffer of the processing unit (11);

◦ else if the presently read audio frame is classified as speech or pause and if the previously read audio frame is not identically classified, regarding it as not proper and forwarding it to an internal speech or pause segment buffer in the processing unit (11), after having performed step (b) on the contents of the speech or pause segment buffer as a segment and emptied it out.

◦ else regarding the presently read audio frame as non-proper and then forwarding the presently read audio frame to an output buffer of the memory unit (12) after having performed step (b) on the contents of the speech or pause segment buffer as a segment and emptied it out.

5. A method according to any one of claims 1 to 4, wherein step (b).1 comprises:

- identifying peaks of a vowel probability function for the speech segment as candidate vowels; and
- determining vowels among candidate vowels as a function of the value of said peaks, and the time between consecutive peaks;
- Identifying intervowel segments as intervals of the speech segment between two successive vowels.

6. A method according to claim 5, wherein the average intervowel distance (T_{avg}) of the speech segment is calculated in step (b).2 either as the average value of durations of the identified intervowel segments, or by using the Kernel Density Estimation algorithm on the durations of the identified intervowel segments.

7. A method according to any one of claims 1 to 6, wherein said nonlinear stretching transfer function is determined is a logarithm function mapping the average intervowel distance (T_{avg}) to the target intervowel distance (T_{target}).

8. A method according to claim 7, wherein determining said nonlinear stretching transfer function in step (b).2 comprises fitting the function $T_{out}(T_{in}) = A \cdot ln(T_{in}) + B$ such that $T_{out}(T_{avg}) = T_{target}$, $T_{out}(T_{min}) = T_{min}$ and $T_{out}(T_{max}) = T_{max}$, wherein T_{min} and T_{max} are given minimum and maximum expected intervowel distance.

9. A method according to any one of claims 1 to 8, wherein the input media content also comprises an input video signal constituted by a sequence of video frames, the input audio signal and the input video signal being synchronized, step (c) comprising generating an output video signal synchronized with the output audio signal by duplicating video frames when needed.

10. A method according to any one of claims 1 to 9, wherein step (b).3 comprises for an intervowel segment:

- determining a target number of samples to be generated for the intervowel segment using the determined stretching transfer function;
- applying stretching process to update the audio frames of the intervowel segment up to reach said target number of samples to be generated.

11. A method according to claim 10, wherein said stretching process comprises for a frame of the intervowel segment:

- calculating a pitch period of the frame;
- for each successive pitch period portion of the frame:

◦ appending a copy of the next pitch period portion to said pitch period portion;

◦ stretching the two pitch periods portion by performing a linear cross-fade between the two pitch periods portion and the copy shifted by a pitch period so as to obtain a three pitch periods portion;

◦ extracting the two first pitch periods portion for output, and sending the remaining pitch period back to the input of the stretching process.

12. A method according to claim 10, wherein said stretching process when repeated comprises for a frame of the intervowel segment:

- calculating a pitch period of the frame;
- for each successive pitch period portion of the frame:

◦ appending a copy of the next pitch period portion to said pitch period portion;

◦ stretching the two pitch periods portion by performing a linear cross-fade between the two pitch periods portion and the copy shifted by a pitch period so as to obtain a three pitch periods portion;

◦ extracting all three pitch periods for output.

13. A method according to any one of claims 11 and 12, wherein the two pitch periods portion is stretched in a first iteration of the starching process only if the two pitch period portion is a silence, else the two

pitch periods portion is stretched in a second iteration of the stretching process only if lag and/or amplitude of an autocorrelation peak of the two pitch periods portion is within a given range.

14. A method according to any one of claims 1 to 9, wherein step (b).3 comprises for an intervowel segment:

- determining a target number of samples to be generated for the intervowel segment using the determined stretching transfer function;
- dividing the intervowel segment into a sequence of periodic, aperiodic or non-stretchable segments, and distributing said target number of samples to be generated between the periodic and aperiodic segments;
- applying a periodic segment stretching process to update the audio frames of the periodic segments and an aperiodic stretching process to update the audio frames of the aperiodic segments up to reach said target number of samples to be generated.

15. A method according to claim 14, wherein said periodic segment stretching process comprises for a periodic segment:

- calculating a pitch period of the periodic segment;
- for each successive pitch period portion of the periodic segment:

◦ extracting said pitch portion for output;
◦ determining a target number of samples to be generated for the pitch period;
◦ stretching said pitch period portion by performing a linear cross-fade between said pitch period portion and the next pitch period portion of the periodic segment so as to obtain a two pitch periods portion.

16. A method according to claim 15, wherein said periodic segment stretching process further comprises, for each successive pitch period portion of the periodic segment, after stretching said pitch portion:

◦ if said target number of samples to be generated for the pitch is exceeded, then further extracting a suitable number of the next samples of the periodic segment for output;
◦ else if said target number of samples to be generated for the pitch period is not reached, then for the next pitch period of the periodic segment:

▪ extracting a suitable number of samples of said next pitch period portion for output;
▪ stretching the last complete pitch period portion in the output by performing a linear cross-fade between said pitch period portion and the next pitch period portion of the periodic segment so as to obtain a two pitch periods portion.

17. A method according to any one of claims 15 and 16, wherein the pitch period portion is stretched only if the lag and/or amplitude of an autocorrelation peak of the pitch period portion is within a given range.

18. A method according to any one of claims 14 to 17, wherein said aperiodic segment stretching process comprises for an aperiodic segment:

- locating zerocrossings of the aperiodic segment;
- for each successive interval extending between a predetermined number of consecutive zerocrossings of the aperiodic segment:

◦ extracting said interval for output;
◦ determining a target number of samples to be generated for the interval;
◦ checking whether said interval is truly aperiodic;
◦ if so, stretching said interval by performing a linear cross-fade between said interval and the next interval extending between said predetermined number of consecutive zerocrossings of the aperiodic segment.

19. A method according to claim 18, wherein said aperiodic segment stretching process further comprises, for each successive interval extending between a predetermined number of consecutive zerocrossings of the aperiodic segment, after stretching said interval:

◦ If said target number of samples to be generated for the interval is exceeded, then further extracting an interval extending between a suitable number of consecutive zerocrossings of the aperiodic segment for output;
◦ Else if said target number of samples to be generated for the interval is not reached, then:

▪ extracting an interval extending between a suitable number of consecutive zerocrossings of the aperiodic segment for output, inferior to said predetermined number;
▪ stretching the last interval extending between said predetermined number of consecutive zerocrossings in the output by performing a linear cross-fade between said interval and the next interval extending between said predetermined number of con-

secutive zerocrossings of the aperiodic segment so as to obtain new samples.

20. A method according to any one of claims 18 and 19, wherein checking whether an interval is truly aperiodic consists in determining if a length of said interval is lower than a threshold.

21. A method according to any one of claims 1 to 20, wherein step (b) also comprises, for each pause segment:

1. calculating an output segment duration T_{out}, based on the segment duration T_{in} and the average intervowel distance T_{avg} and target intervowel distance T_{target} from the previous speech segment;
2. stretching the pause segment using the determined output duration, so as to generate updated audio frames of the pause segment.

22. An equipment (1) comprising a processing unit (11) configured to perform:

- a module for receiving input media content comprising at least an input audio signal constituted by a sequence of audio frames;
- a module for classifying of the audio frames as speech, non-speech, or pause, so as to divide said audio signal of the media content into speech segments bounded by non-speech segments;
- A module for dividing each speech segment into a sequence of intervowel segments;
- A module for calculating an average intervowel distance (T_{avg}) of each divided speech segment, and for determining a non-linear stretching transfer function mapping an input intervowel distance (T_{in}) to an output intervowel distance (T_{out}) as a function of said average intervowel distance (T_{avg}) and a given target intervowel distance (T_{target});
- A module for stretching each intervowel segment using the determined stretching transfer function so as to generate updated audio frames of the intervowel segment;
- A module for generating an output media content comprising as output signal the input audio signal wherein for each intervowel segment of each speech segment the corresponding audio frames have been replaced by the updated audio frames.

23. An equipment according to claim 22, comprising a display (2) for displaying the generated media content.

24. A computer program product, comprising code instructions for executing a method according to any one of claims 1 to 21 for slowing down speech in an input media content.

25. A computer-readable medium, on which is stored a computer program product comprising code instructions for executing a method according to any one of claims 1 to 21 for slowing down speech in an input media content.

FIG. 1

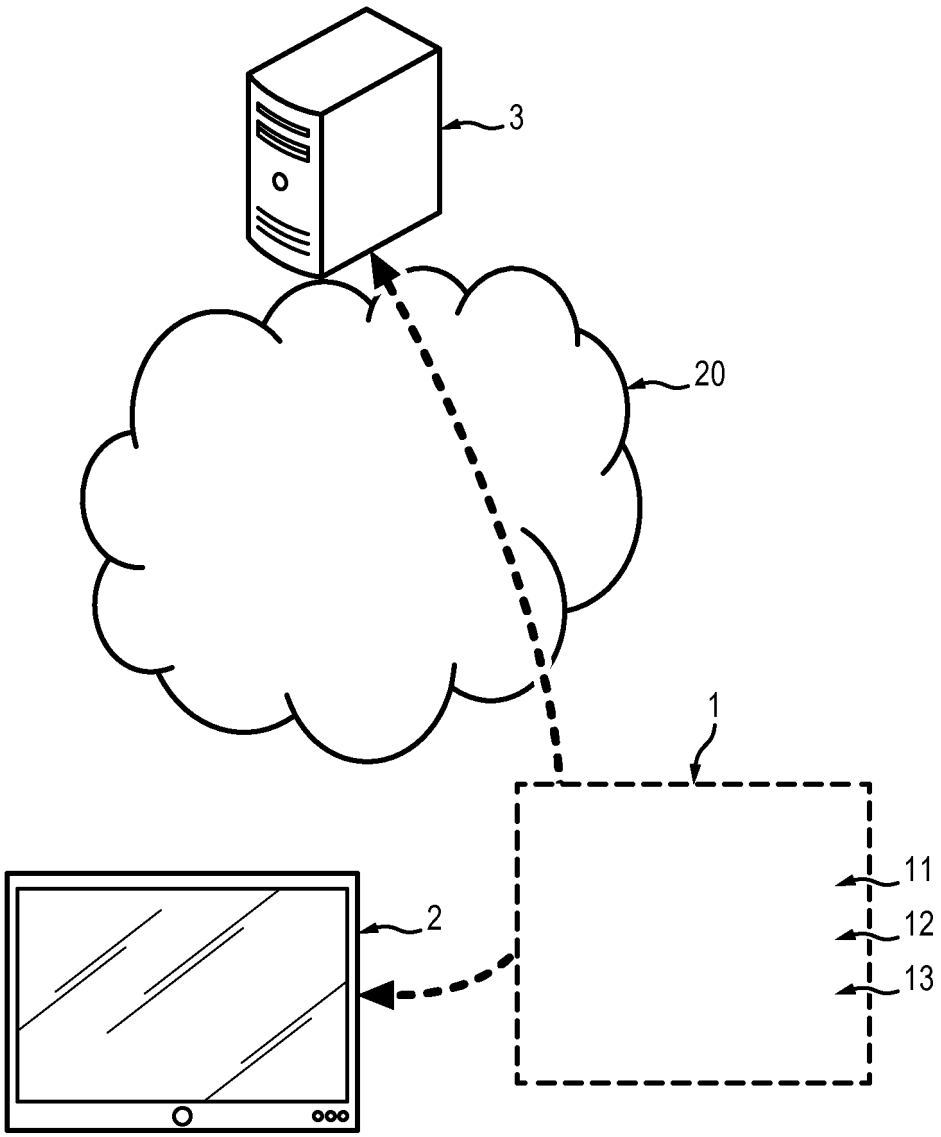


FIG. 2

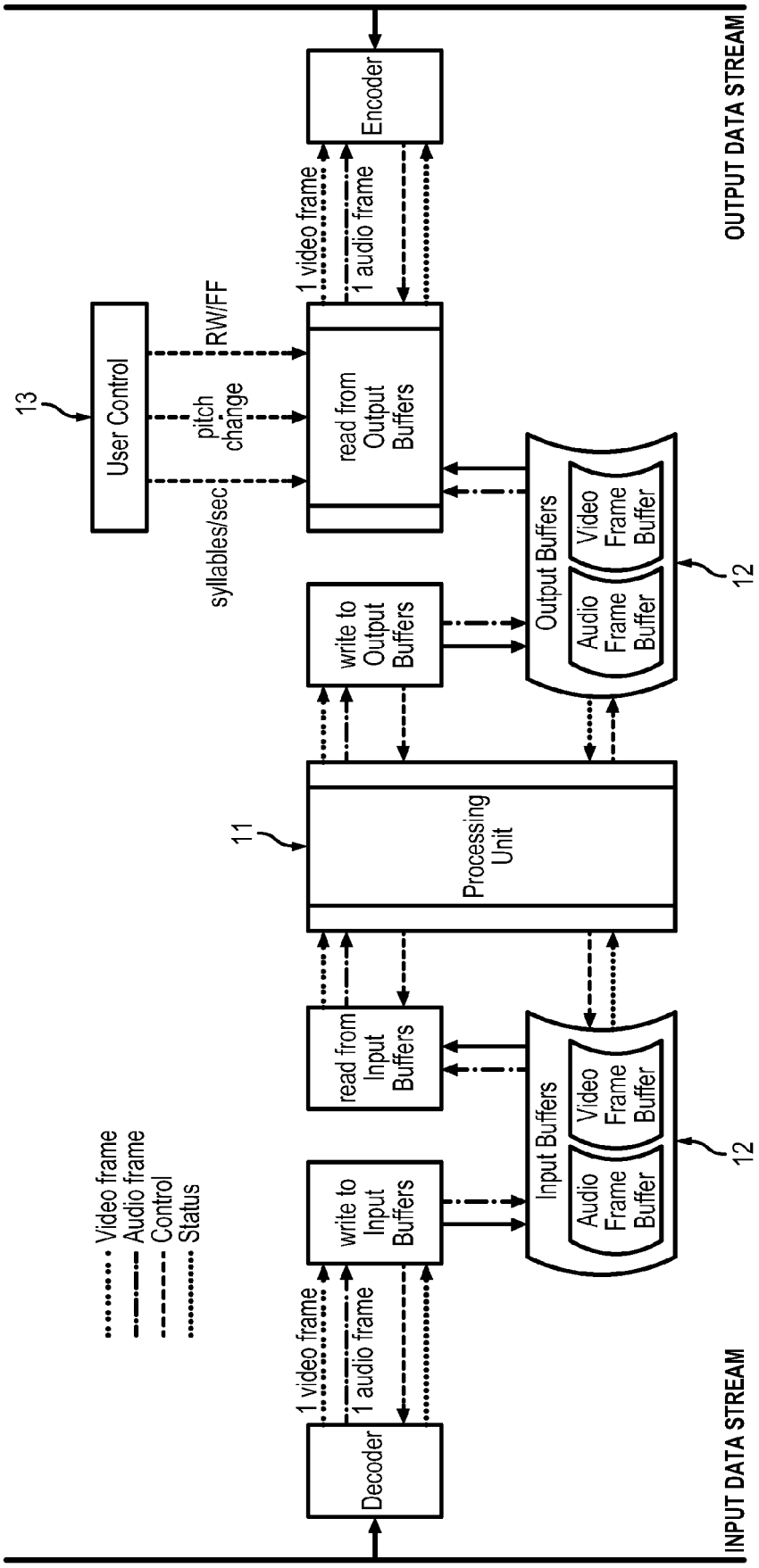

OUTPUT DATA STREAM
Encoder
1 video frame
1 audio frame
13
User Control
RW/FF
pitch change
syllables/sec
read from Output Buffers
write to Output Buffers
Output Buffers
Video Frame Buffer
Audio Frame Buffer
12
11
Processing Unit
read from Input Buffers
write to Input Buffers
Input Buffers
Video Frame Buffer
Audio Frame Buffer
12
Video frame
Audio frame
Control
Status
1 video frame
1 audio frame
Decoder
INPUT DATA STREAM

FIG. 3

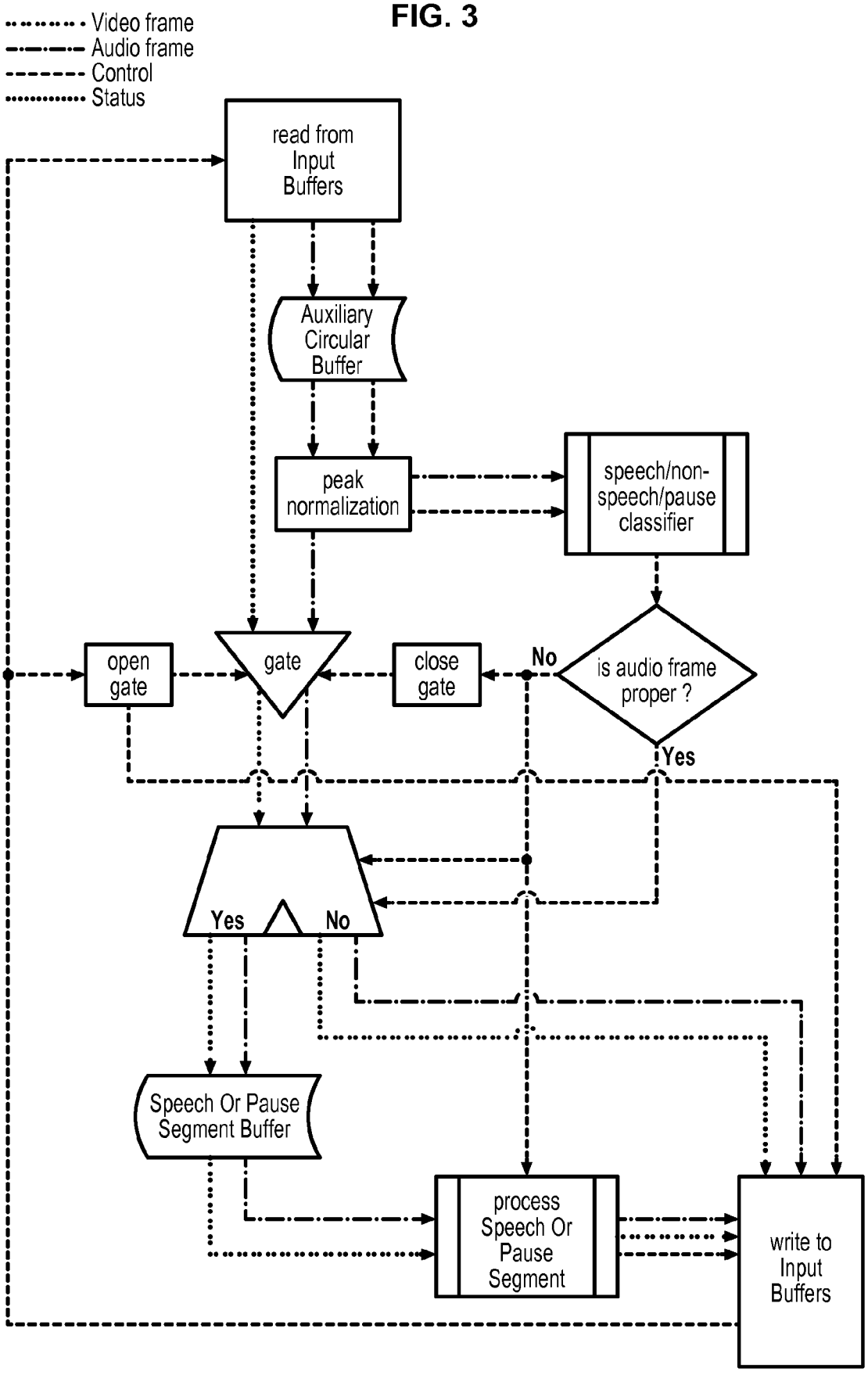

Video frame
Audio frame
Control
Status
read from Input Buffers
Auxiliary Circular Buffer
peak normalization
speech/non-speech/pause classifier
open gate
gate
close gate
No
is audio frame proper ?
Yes
Yes
No
Speech Or Pause Segment Buffer
process Speech Or Pause Segment
write to Input Buffers

FIG. 4

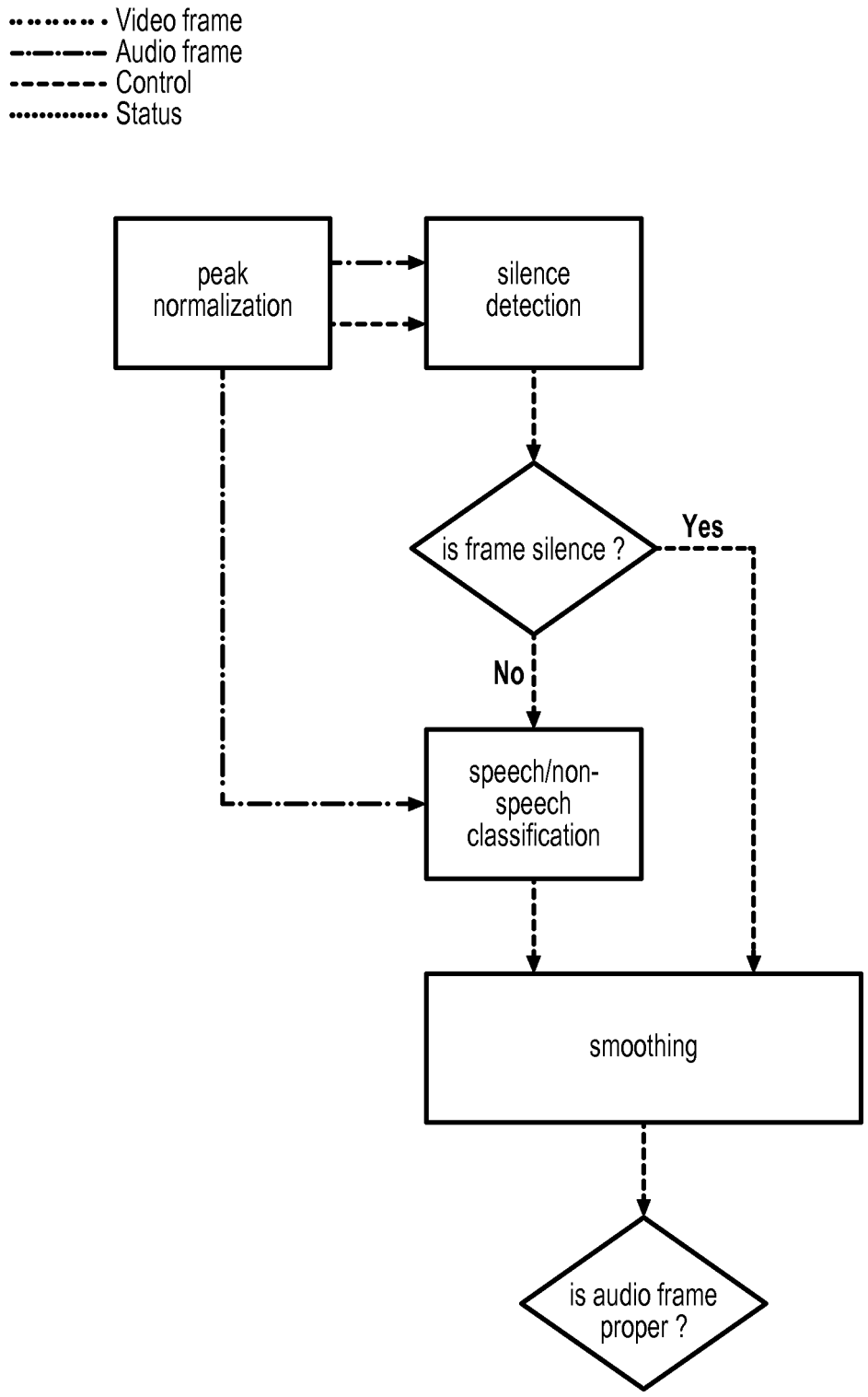

Video frame
Audio frame
Control
Status
peak normalization
silence detection
is frame silence ?
Yes
No
speech/non-speech classification
smoothing
is audio frame proper ?

FIG. 5

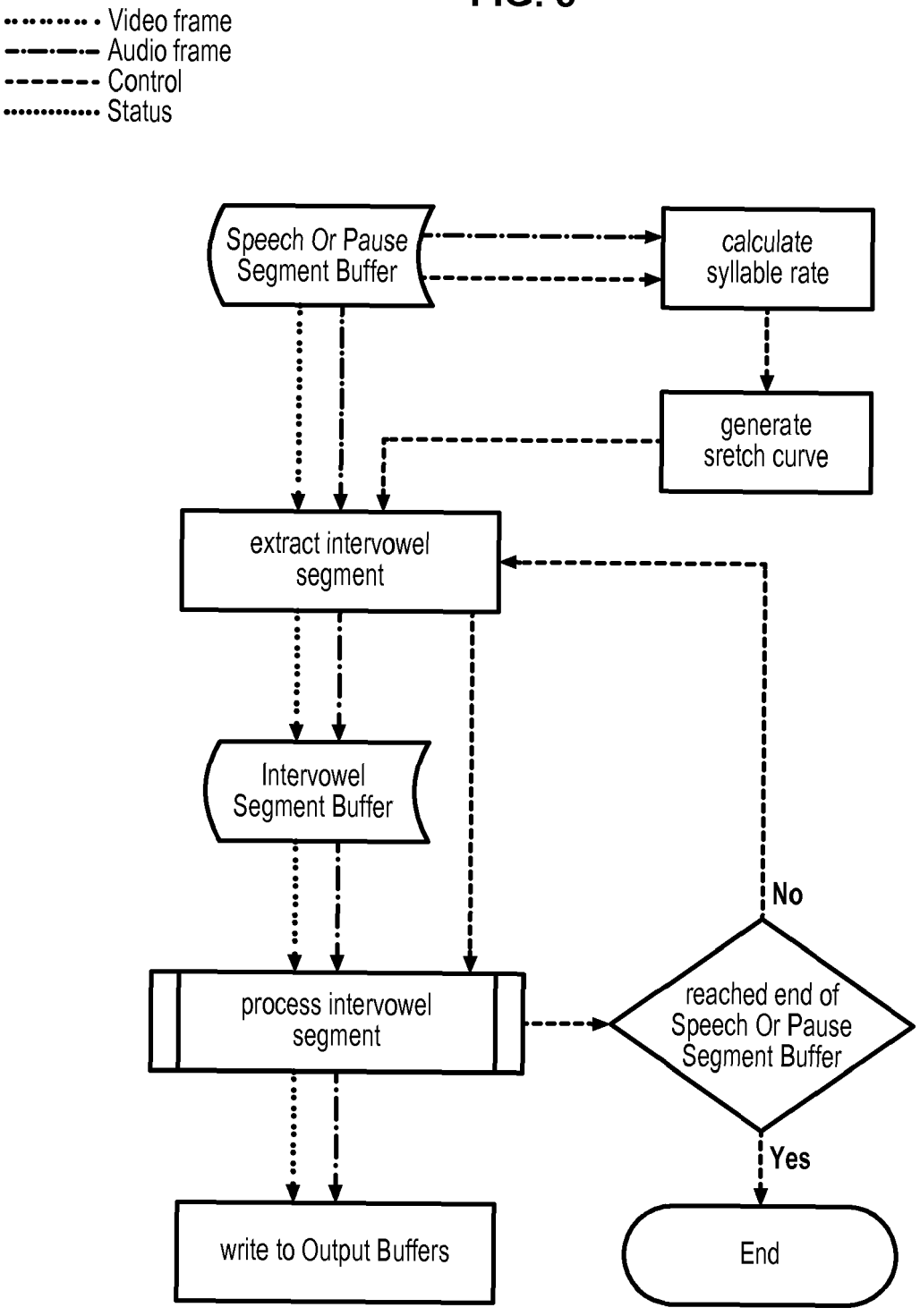

Video frame
Audio frame
Control
Status
Speech Or Pause Segment Buffer
calculate syllable rate
generate sretch curve
extract intervowel segment
Intervowel Segment Buffer
process intervowel segment
reached end of Speech Or Pause Segment Buffer
No
Yes
write to Output Buffers
End

FIG. 6

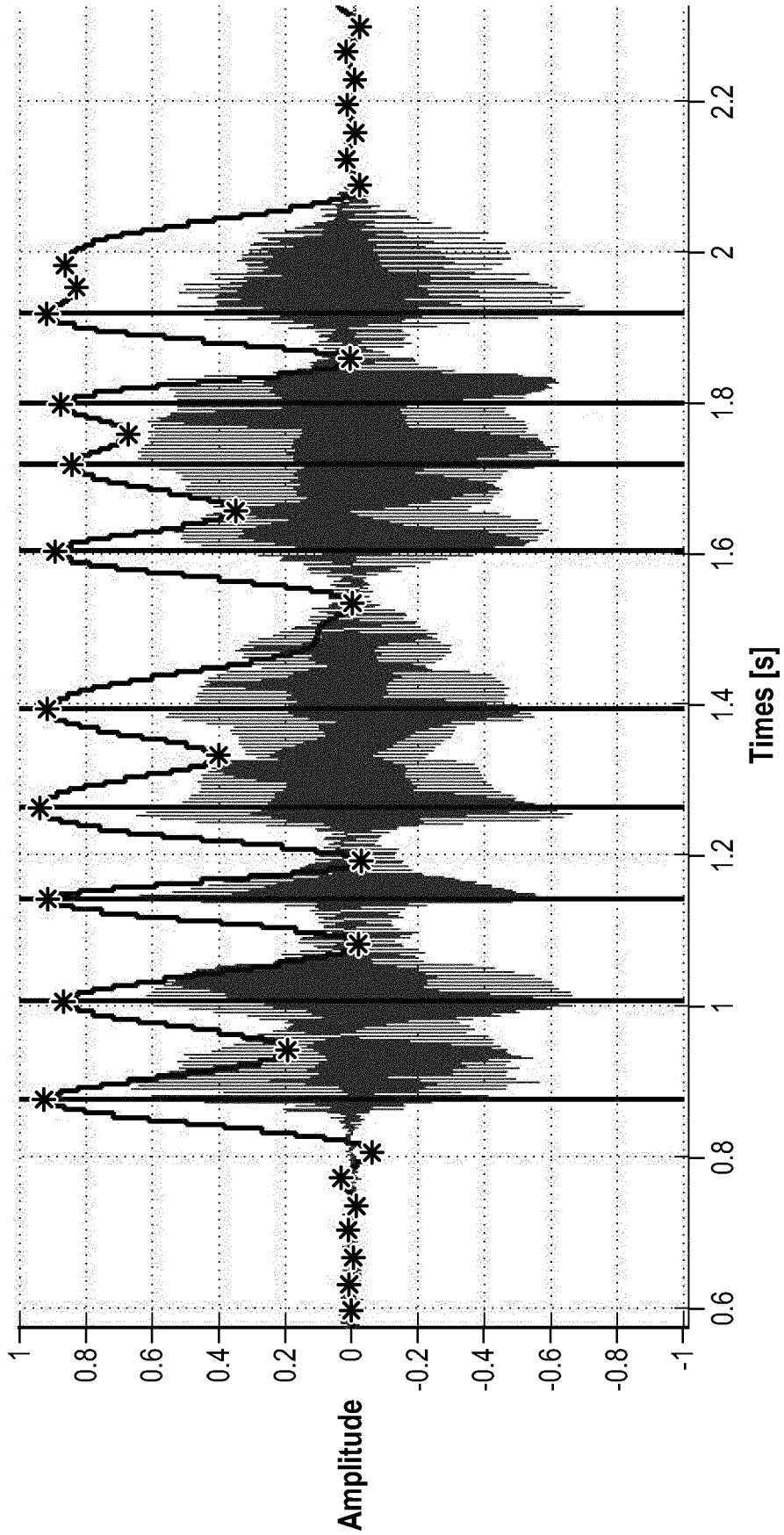

Amplitude
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
Times [s]
0.6
0.8
1
1.2
1.4
1.6
1.8
2
2.2

FIG. 7

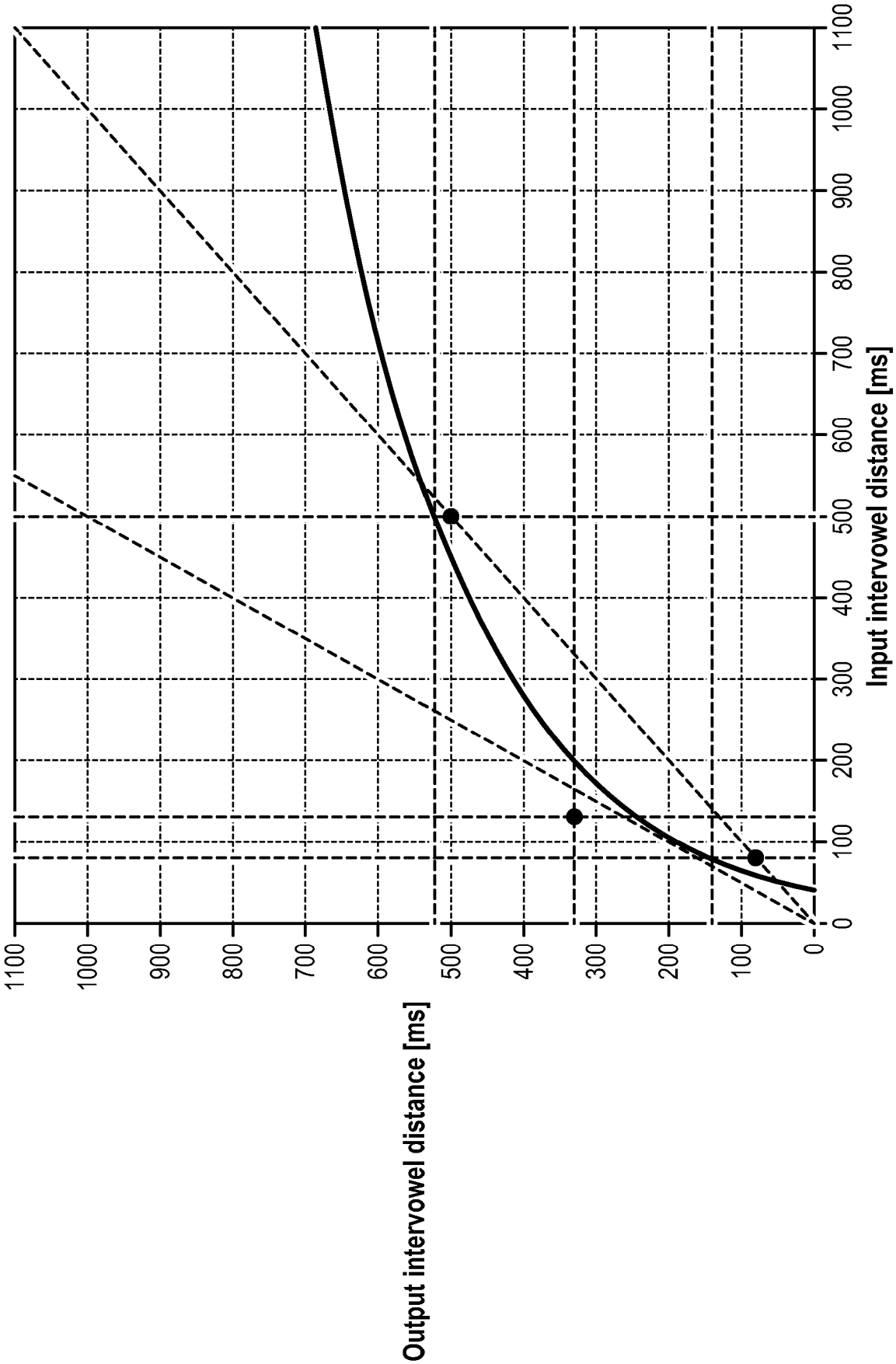

Output intervowel distance [ms]
1100
1000
900
800
700
600
500
400
300
200
100
0
Input intervowel distance [ms]
0
100
200
300
400
500
600
700
800
900
1000
1100

FIG. 8a

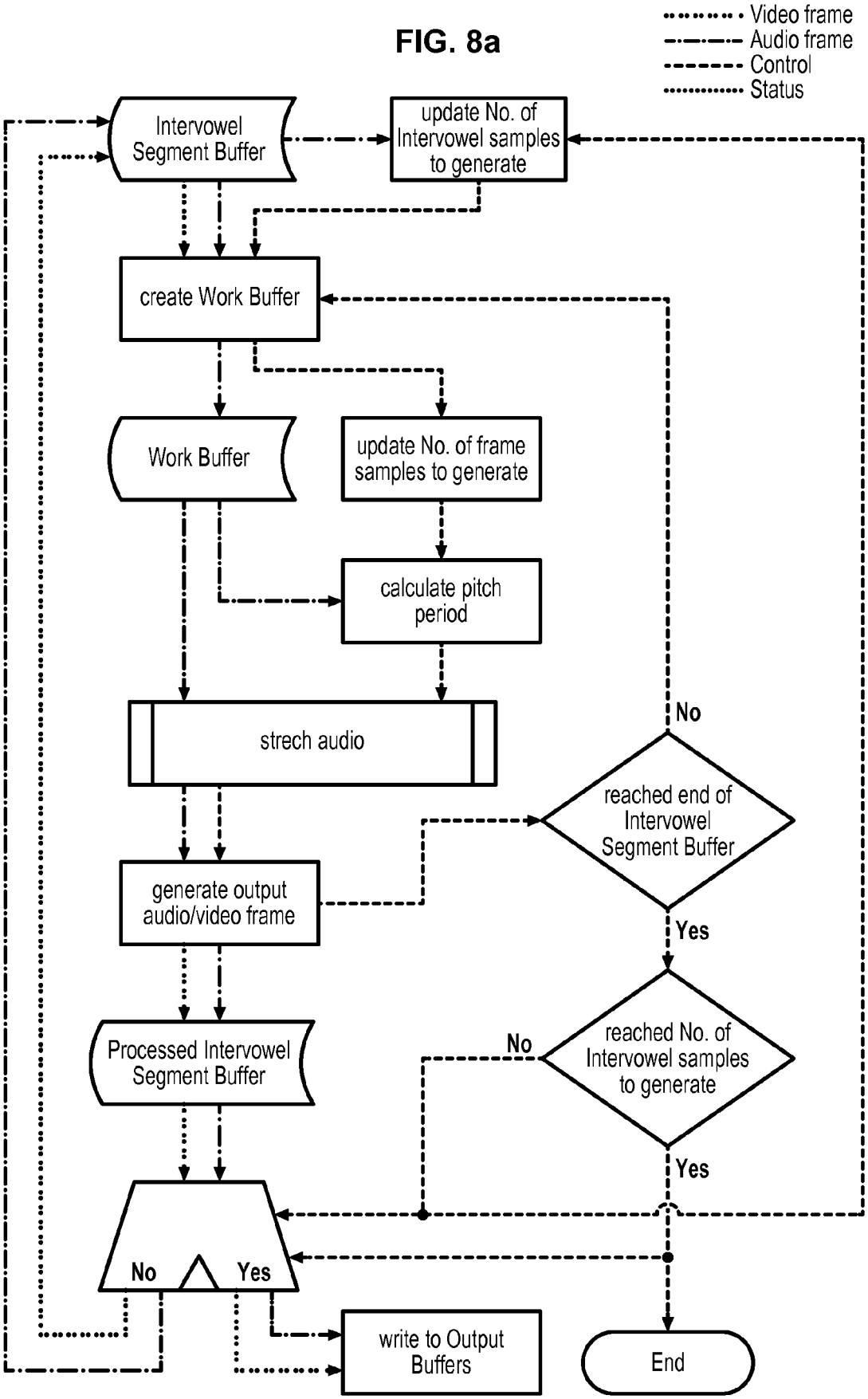

Video frame
Audio frame
Control
Status
Intervowel Segment Buffer
update No. of Intervowel samples to generate
create Work Buffer
Work Buffer
update No. of frame samples to generate
calculate pitch period
strech audio
reached end of Intervowel Segment Buffer
No
Yes
generate output audio/video frame
Processed Intervowel Segment Buffer
reached No. of Intervowel samples to generate
No
Yes
No
Yes
write to Output Buffers
End

FIG. 8b

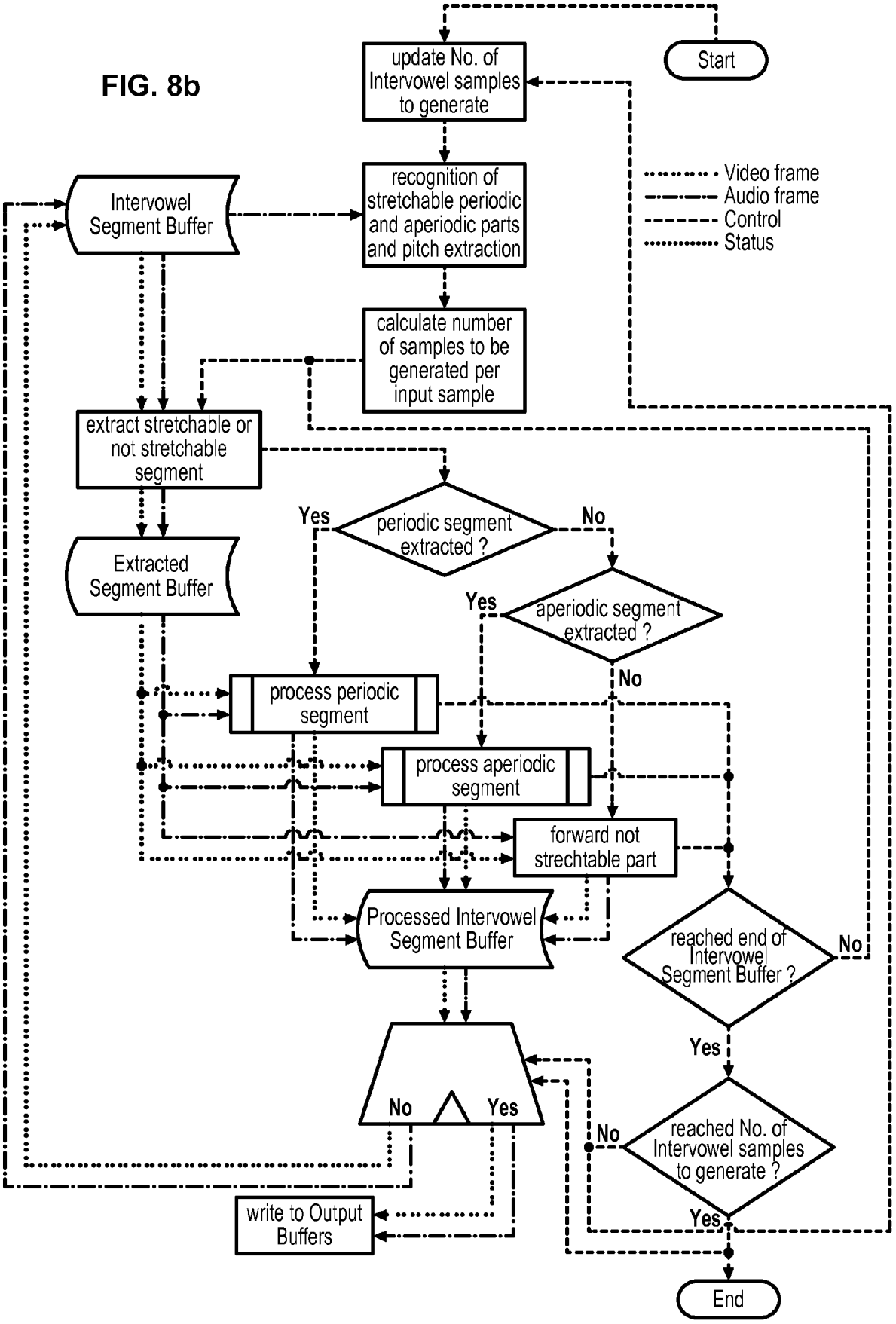

Start
update No. of Intervowel samples to generate
recognition of stretchable periodic and aperiodic parts and pitch extraction
calculate number of samples to be generated per input sample
Intervowel Segment Buffer
extract stretchable or not stretchable segment
Extracted Segment Buffer
Video frame
Audio frame
Control
Status
Yes
periodic segment extracted ?
No
Yes
aperiodic segment extracted ?
No
process periodic segment
process aperiodic segment
forward not strechtable part
Processed Intervowel Segment Buffer
reached end of Intervowel Segment Buffer ?
No
Yes
No
Yes
reached No. of Intervowel samples to generate ?
No
Yes
write to Output Buffers
End

FIG. 9a

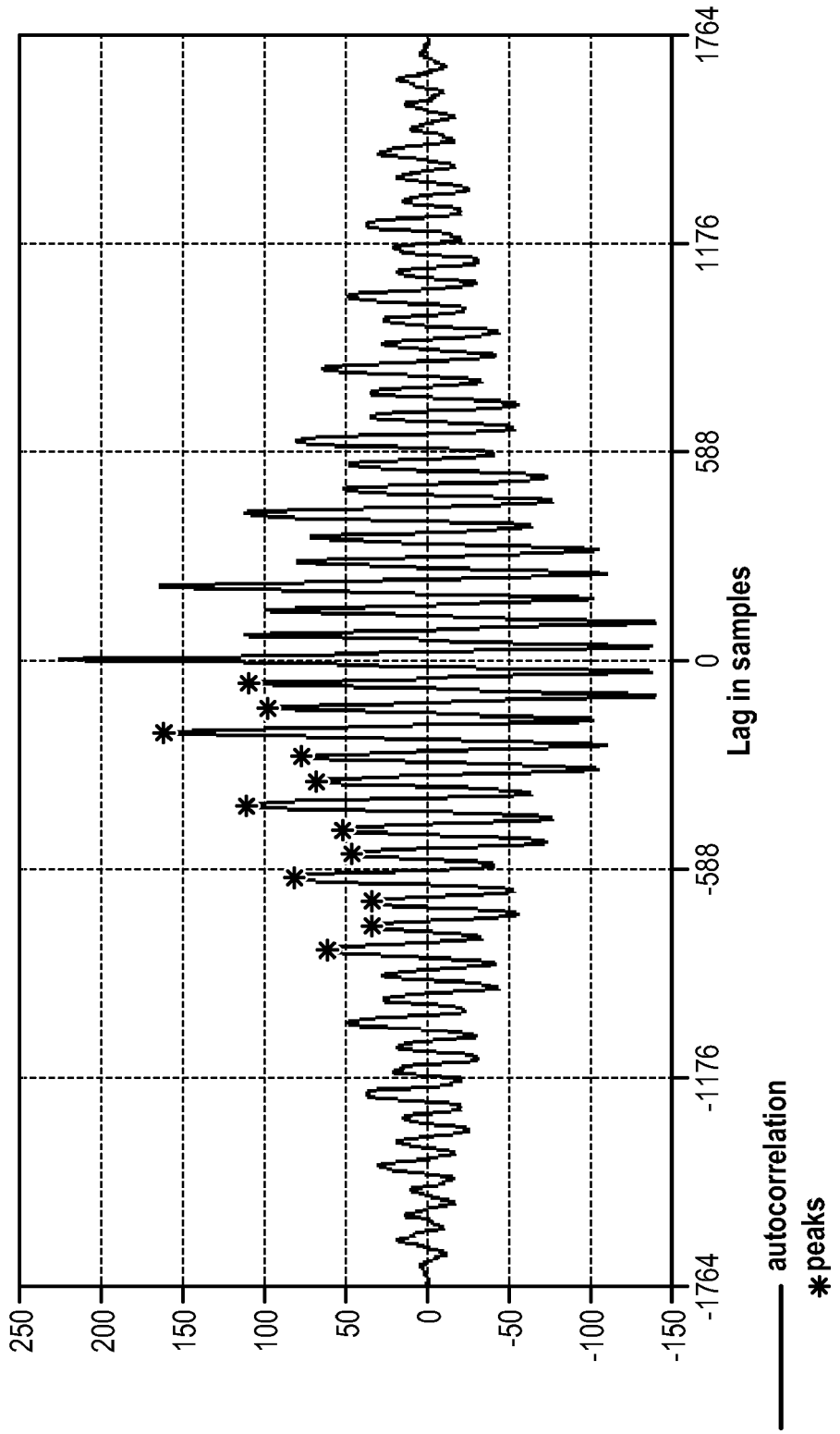

250
200
150
100
50
0
-50
-100
-150
-1764
-1176
-588
0
588
1176
1764
Lag in samples
autocorrelation
peaks

FIG. 9b

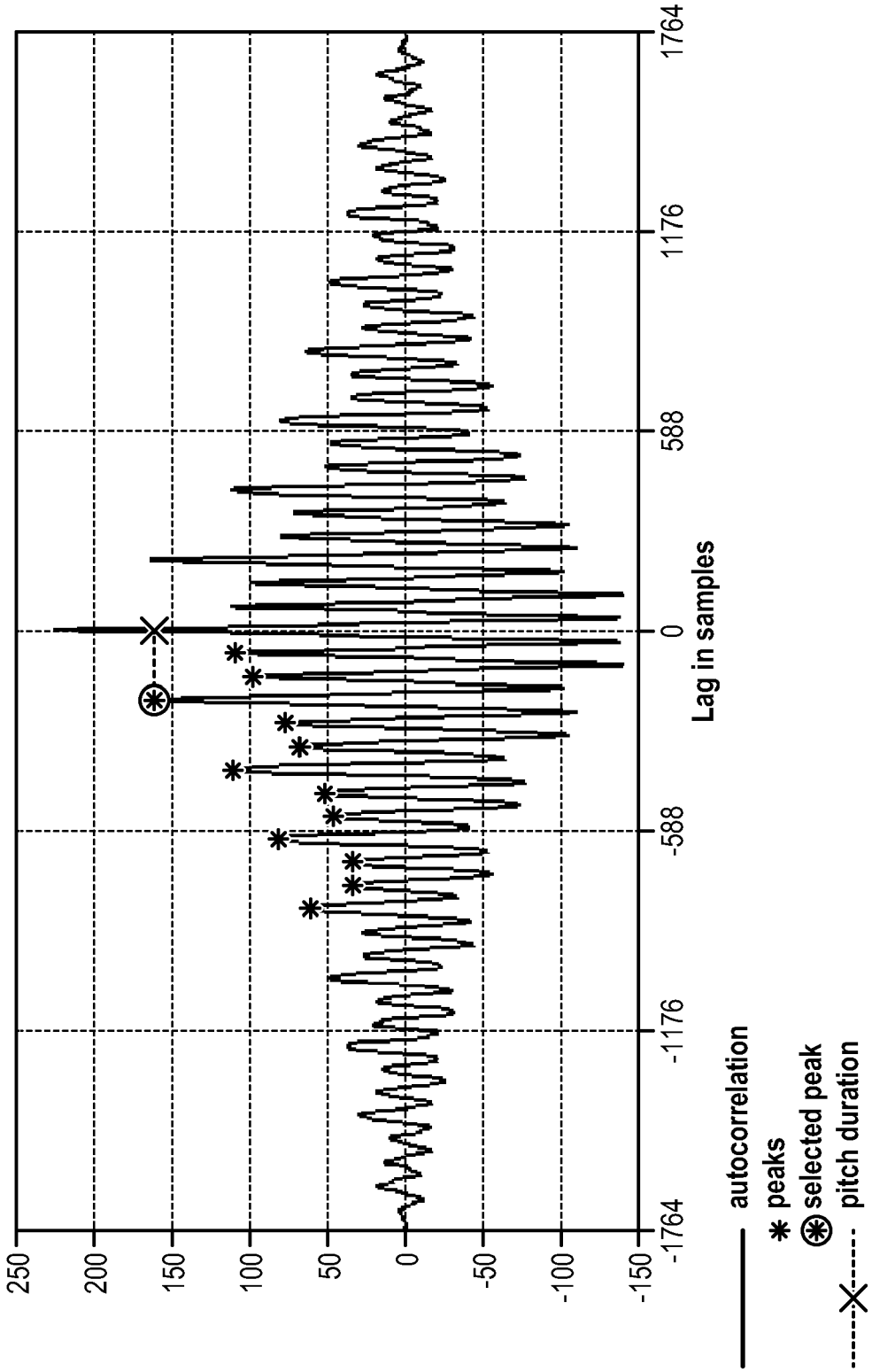

250
200
150
100
50
0
-50
-100
-150
-1764
-1176
-588
0
588
1176
1764
Lag in samples
autocorrelation
peaks
selected peak
pitch duration

FIG. 10

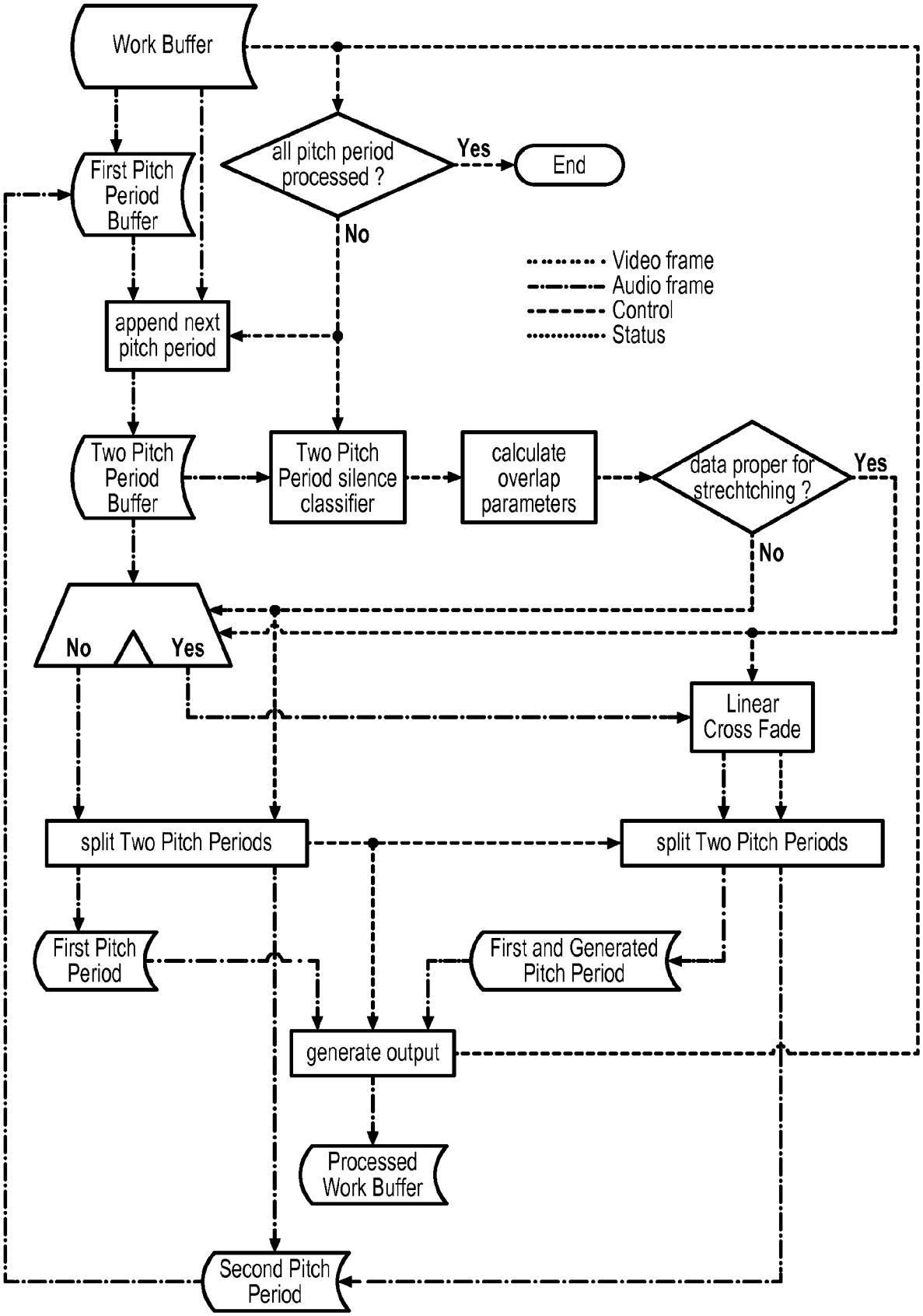
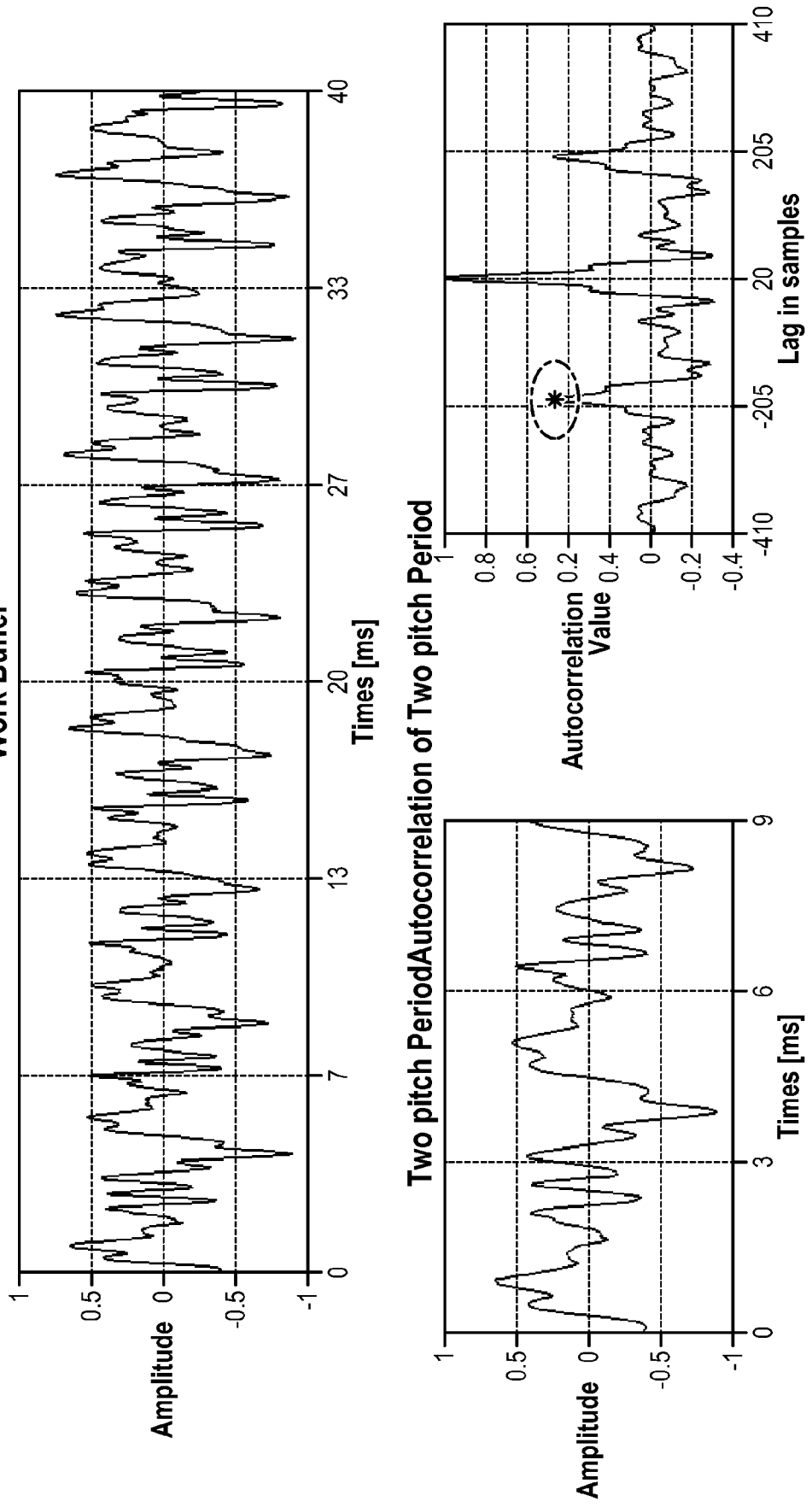

Work Buffer
all pitch period processed ?
Yes
End
No
First Pitch Period Buffer
Video frame
Audio frame
Control
Status
append next pitch period
Two Pitch Period Buffer
Two Pitch Period silence classifier
calculate overlap parameters
data proper for strechtching ?
Yes
No
No
Yes
Linear Cross Fade
split Two Pitch Periods
split Two Pitch Periods
First Pitch Period
First and Generated Pitch Period
generate output
Processed Work Buffer
Second Pitch Period

FIG. 11a


Work Buffer
Amplitude
Times [ms]
Two pitch PeriodAutocorrelation of Two pitch Period
Autocorrelation Value
Lag in samples

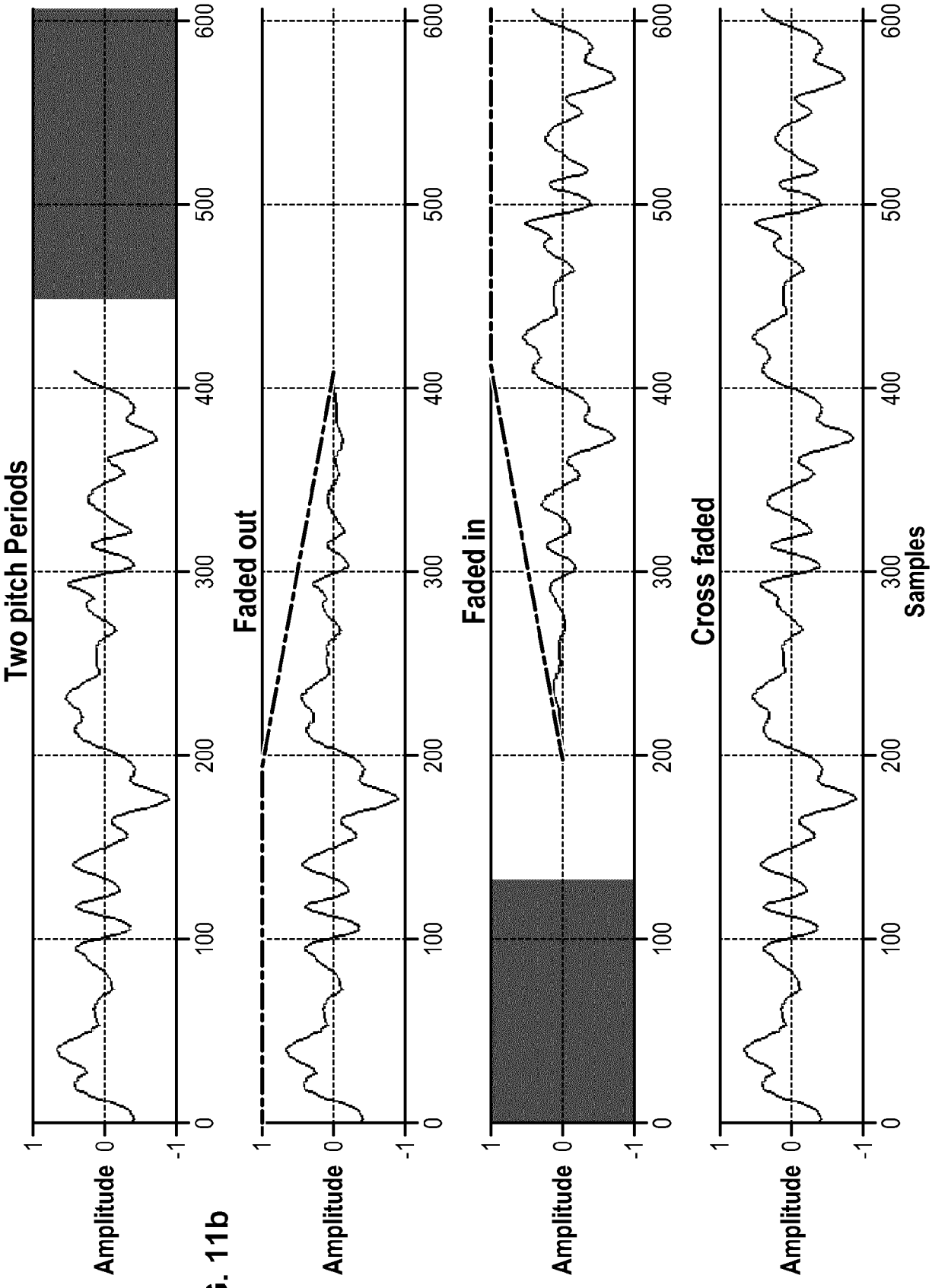

Two pitch Periods
Faded out
Faded in
Cross faded
Amplitude
1
0
-1
0
100
200
300
400
500
600
Samples

FIG. 11b

FIG. 12

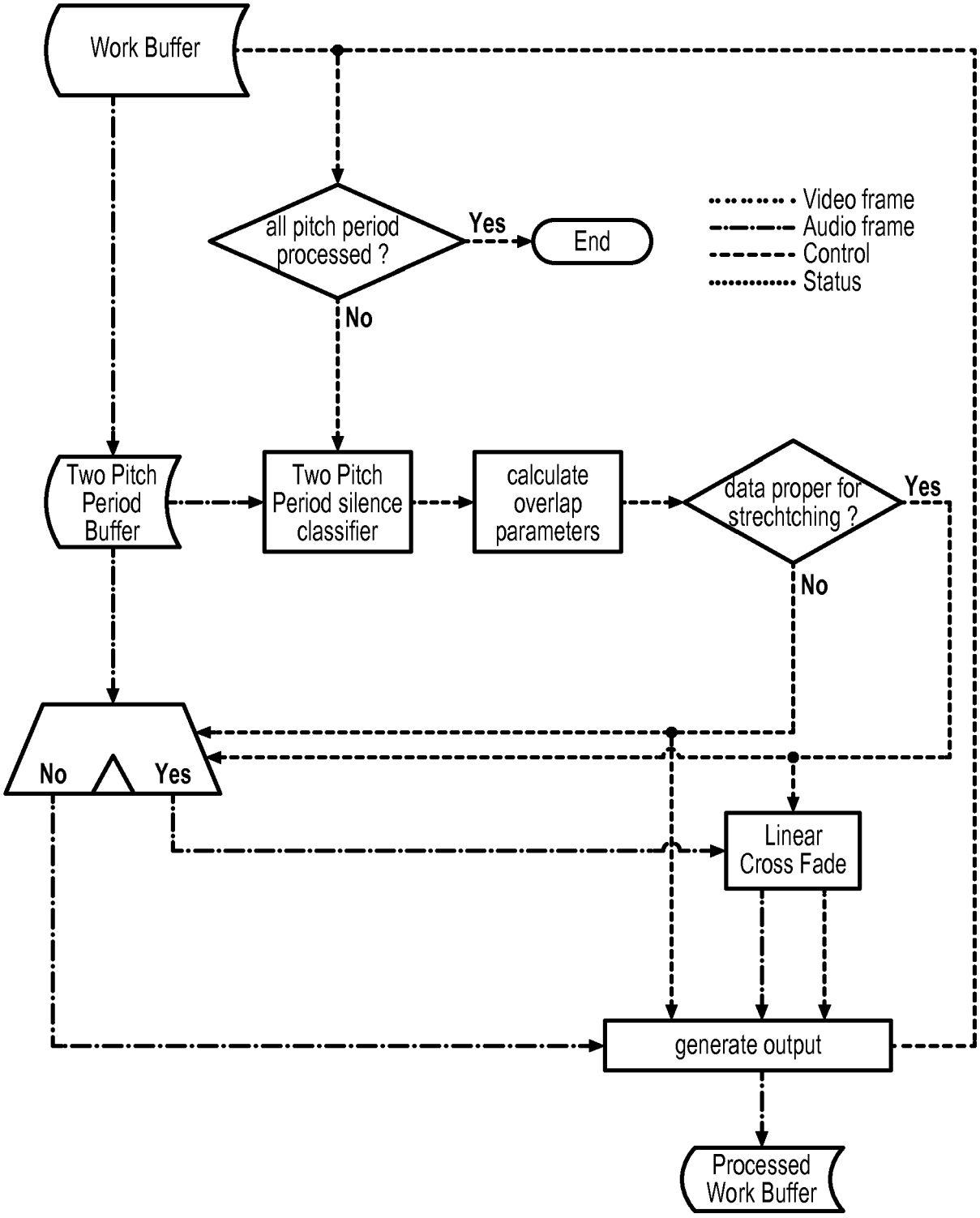

Work Buffer
all pitch period processed ?
Yes
End
No
Video frame
Audio frame
Control
Status
Two Pitch Period Buffer
Two Pitch Period silence classifier
calculate overlap parameters
data proper for strechtching ?
Yes
No
No
Yes
Linear Cross Fade
generate output
Processed Work Buffer

FIG. 13a

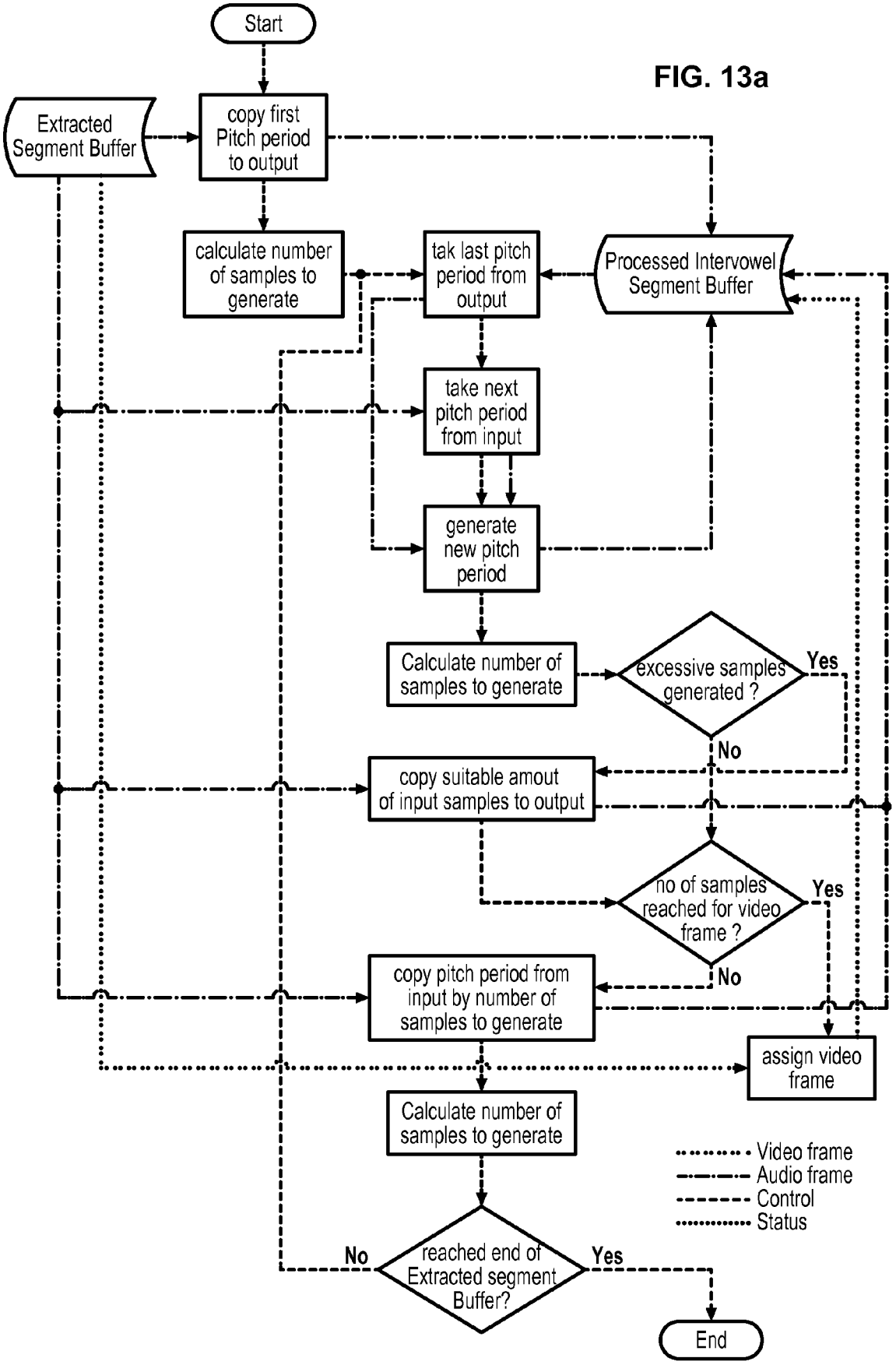

Start
Extracted Segment Buffer
copy first Pitch period to output
calculate number of samples to generate
tak last pitch period from output
Processed Intervowel Segment Buffer
take next pitch period from input
generate new pitch period
Calculate number of samples to generate
excessive samples generated ?
Yes
No
copy suitable amout of input samples to output
no of samples reached for video frame ?
Yes
No
copy pitch period from input by number of samples to generate
assign video frame
Calculate number of samples to generate
reached end of Extracted segment Buffer?
No
Yes
End
Video frame
Audio frame
Control
Status

FIG. 13b

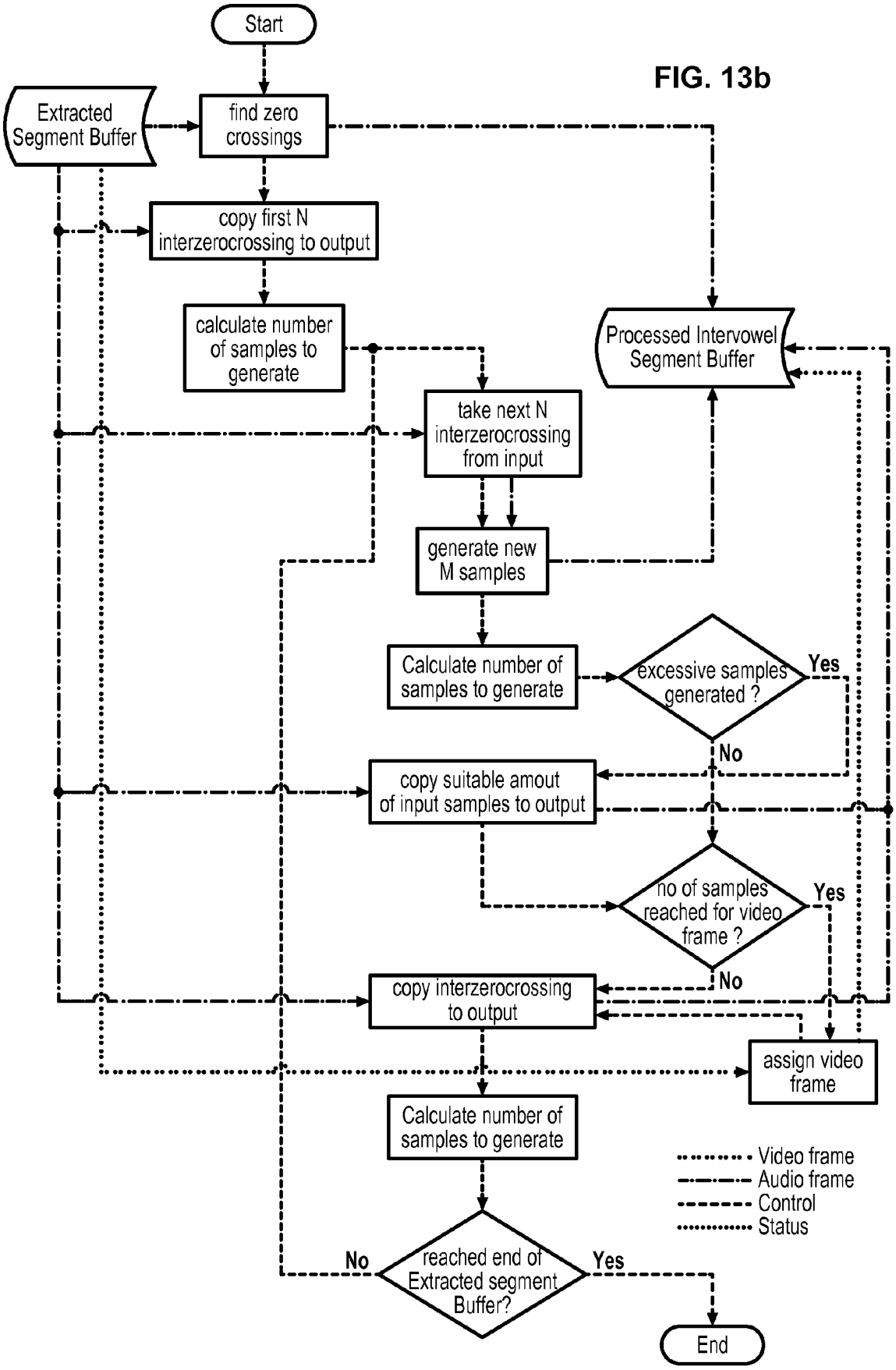

Start
Extracted Segment Buffer
find zero crossings
copy first N interzerocrossing to output
calculate number of samples to generate
Processed Intervowel Segment Buffer
take next N interzerocrossing from input
generate new M samples
Calculate number of samples to generate
excessive samples generated ?
Yes
No
copy suitable amout of input samples to output
no of samples reached for video frame ?
Yes
No
copy interzerocrossing to output
assign video frame
Calculate number of samples to generate
reached end of Extracted segment Buffer?
No
Yes
End
Video frame
Audio frame
Control
Status

FIG. 14a

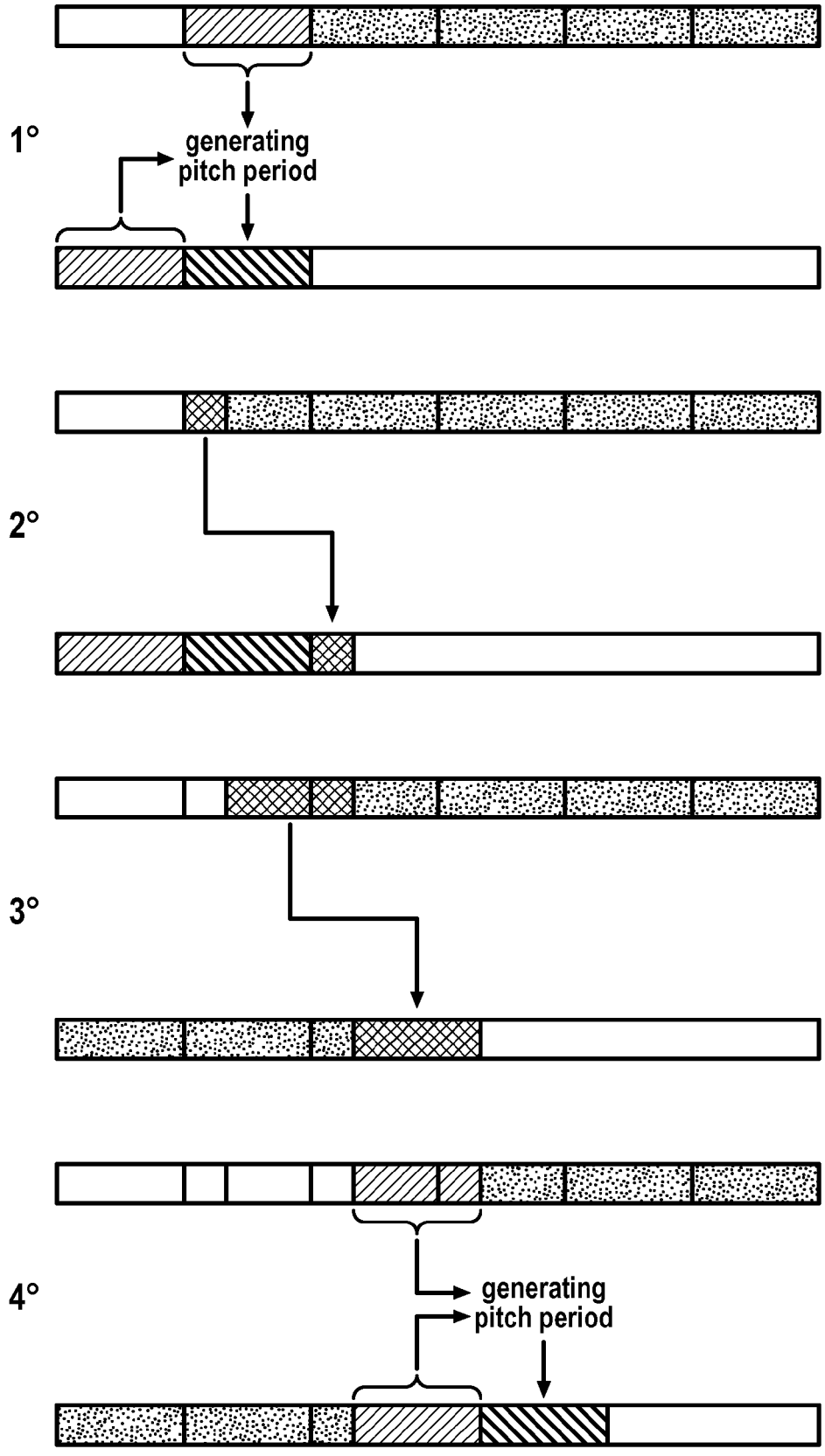
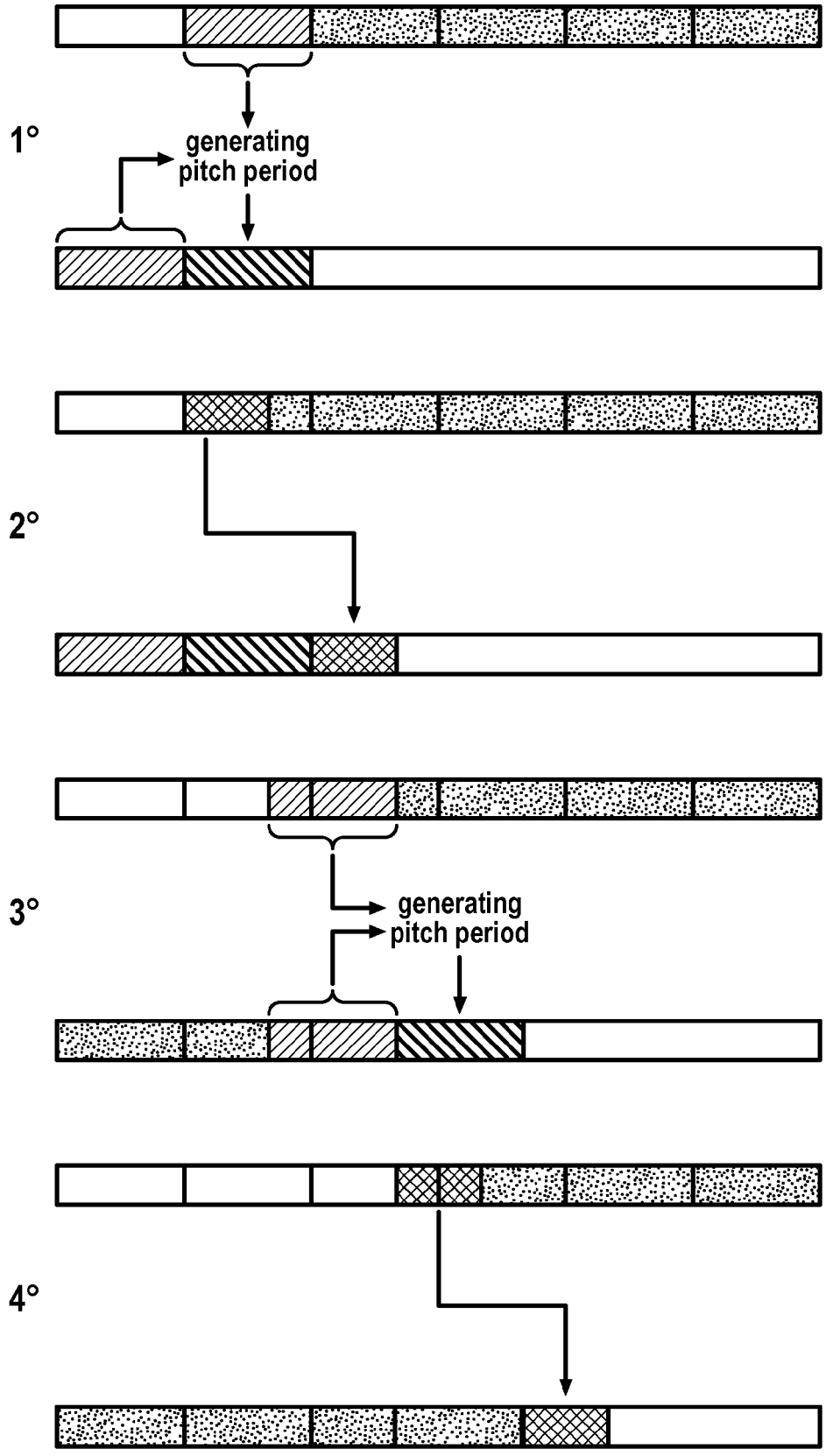


FIG. 14b



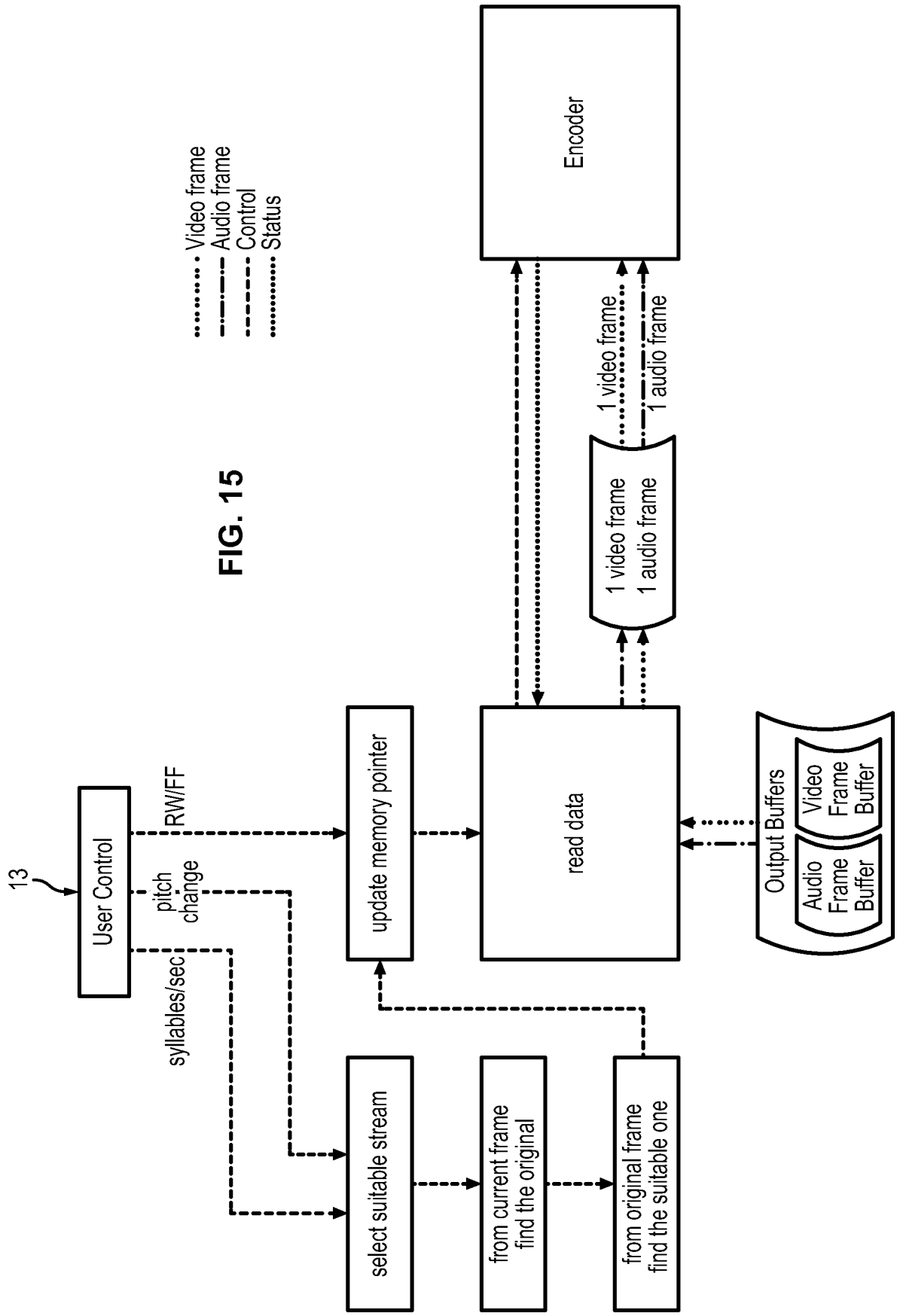
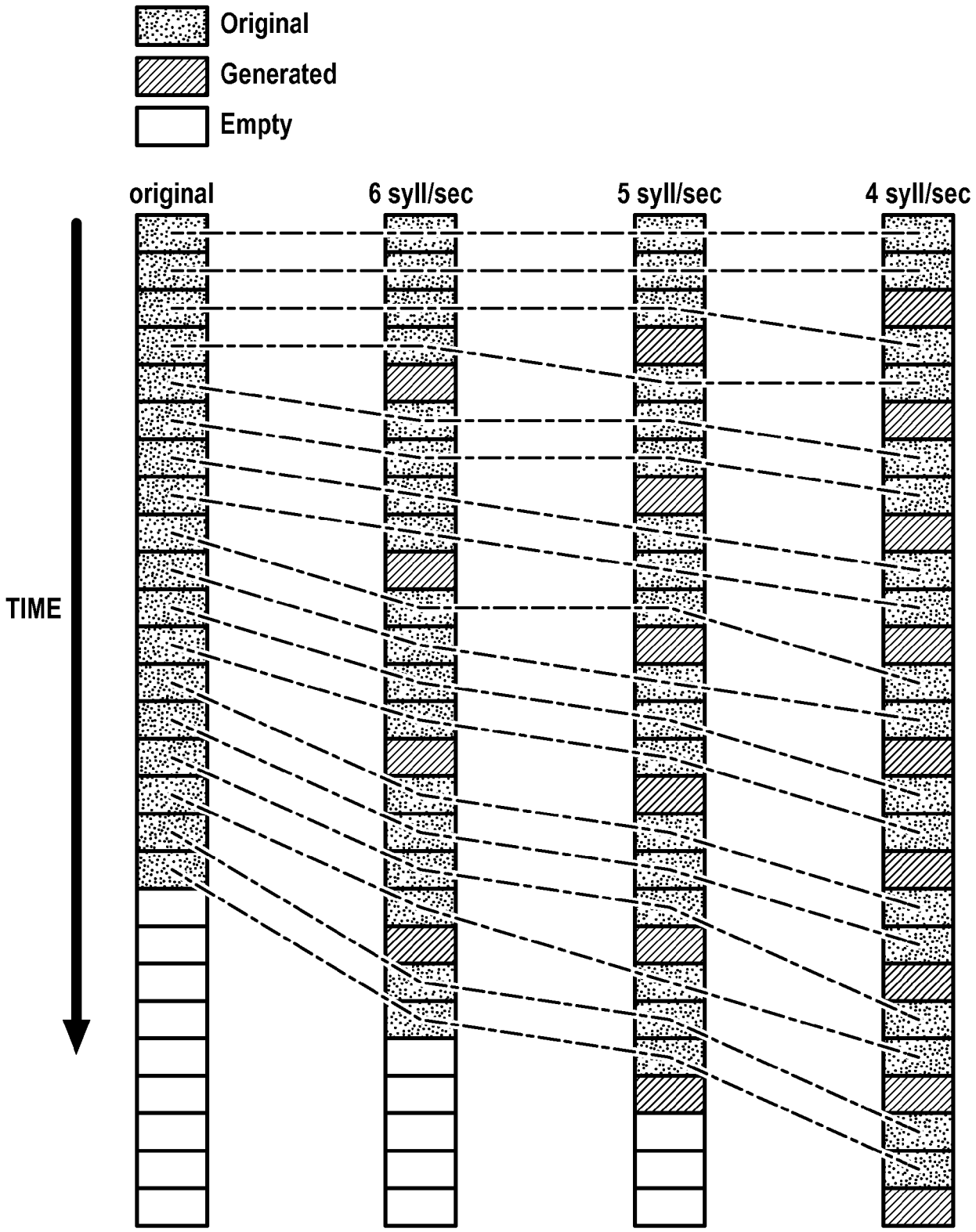

Video frame
Audio frame
Control
Status
Encoder
1 video frame
1 audio frame
1 video frame
1 audio frame
13
User Control
RW/FF
pitch change
syllables/sec
update memory pointer
read data
Output Buffers
Video Frame Buffer
Audio Frame Buffer
select suitable stream
from current frame find the original
from original frame find the suitable one

FIG. 15

FIG. 16


Original
Generated
Empty
original
6 syll/sec
5 syll/sec
4 syll/sec
TIME

Europäisches Patentamt
European Patent Office
Office européen des brevets

EUROPEAN SEARCH REPORT

Application Number
EP 16 30 6550

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	WO 97/46999 A1 (INTERVAL RESEARCH CORP [US]) 11 December 1997 (1997-12-11)	1-3, 5-22,24, 25	INV. G10L21/04
Y	* page 11, line 25 - page 12, line 14; figures 3, 4 * * page 14, line 27 - page 15, line 6 * * page 7, line 4 - page 9, line 4 * * page 16, line 8 - line 23 *	4,23	
X	US 2011/004468 A1 (FUSAKAWA KAZUE [JP] ET AL) 6 January 2011 (2011-01-06) * figures 1, 5-9, 19, 25, 26 * * paragraph [0011] - paragraph [0012] * * paragraph [0117] * * paragraph [0251] - paragraph [0256] *	1-3,5-8, 10-22, 24,25	
X	US 7 065 485 B1 (CHONG-WHITE NICOLA R [US] ET AL) 20 June 2006 (2006-06-20) * figures 1,7 * * column 5, line 56 - column 6, line 19 * * column 4, line 53 - column 5, line 3 * * column 7, line 48 - column 8, line 54 *	1-3,5-8, 10-22, 24,25	TECHNICAL FIELDS SEARCHED (IPC) G10L
Y	US 2004/267524 A1 (BOILLOT MARC ANDRE [US] ET AL) 30 December 2004 (2004-12-30)	4,23	
A	* paragraph [0091] * * paragraph [0052] *	10-21	

The present search report has been drawn up for all claims

Place of search	Date of completion of the search	Examiner
Munich	2 June 2017	Müller, Achim

CATEGORY OF CITED DOCUMENTS

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

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

& : member of the same patent family, corresponding document

EPO FORM 1503 03.82 (P04C01)

ANNEX TO THE EUROPEAN SEARCH REPORT ON EUROPEAN PATENT APPLICATION NO. EP 16 30 6550

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

02-06-2017

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
WO 9746999 A1	11-12-1997	AU 719955 B2	18-05-2000
		CA 2257298 A1	11-12-1997
		EP 0978119 A1	09-02-2000
		JP 2000511651 A	05-09-2000
		US 5828994 A	27-10-1998
		WO 9746999 A1	11-12-1997
US 2011004468 A1	06-01-2011	CN 101939784 A	05-01-2011
		EP 2383732 A1	02-11-2011
		JP 5870309 B2	24-02-2016
		JP 2014194554 A	09-10-2014
		US 2011004468 A1	06-01-2011
		WO 2010087171 A1	05-08-2010
US 7065485 B1	20-06-2006	NONE	
US 2004267524 A1	30-12-2004	NONE	

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 7853447 B **[0010]**
- US 6484137 B **[0011]**
- US 7412379 B **[0012]**

Non-patent literature cited in the description

- **A. KUPRYJANOW ; A. CZYZEWSKI.** *Methods of Improving Speech Intelligibility for Listeners with Hearing Resolution Deficit,* 2012 **[0014]**
- A pitch extraction algorithm tuned for automatic speech recognition. **GHAHREMANI ; PEGAH ; BAGHER BABA ALI ; DANIEL POVEY ; KORBINIAN RIEDHAMMER ; JAN TRMAL ; SANJEEV KHUDANPUR.** IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2014, 2494-2498 **[0107]**