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

(11) **EP 3 447 767 A1**

(12) **EUROPEAN PATENT APPLICATION**

(43) Date of publication:
**27.02.2019 Bulletin 2019/09**

(51) Int Cl.:
***G10L 21/04*** *(2013.01)* ***G10L 19/02*** *(2013.01)*

(21) Application number: **17187265.8**

(22) Date of filing: **22.08.2017**

(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: **Österreichische Akademie der Wissenschaften**
**1010 Wien (AT)**

(72) Inventors:
- **Prusa, Zdenek**
  **37364 Dynin (CZ)**
- **Holighaus, Nicki**
  **2344 Maria Enzersdorf (AT)**
- **Søndergaard, Peter L.**
  **2300 Kopenhagen (DK)**

(74) Representative: **Patentanwaltskanzlei Matschnig & Forsthuber OG**
**Biberstraße 22**
**Postfach 36**
**1010 Wien (AT)**

# (54) METHOD FOR PHASE CORRECTION IN A PHASE VOCODER AND DEVICE

(57) A method for phase correction in a phase vocoder, comprising the steps of:
A. Receiving an audio input signal (10),
B. Determining at least a first, a second and a third time frame of said audio input signal (10) and obtaining a first, a second and a third time frame index parameter (20, 21, 22),
C. Calculating a first, a second and a third signal representation value set (30, 31, 32), for said audio input signal (10) related to each of said first, said second and said third time frame index parameter (20, 21, 22), and the calculation of each of said first, said second and said third signal representation value set (30, 31, 32) comprises signal representation parameters derived from said audio input signal (10) using a windowing function and a short-time Fourier transformation,
D. Calculating a phase derivative value set (50) using finite differences,
E1. Adjusting said phase derivative value set (50) and obtaining an adjusted phase derivative value set (60),
E2. Integrating said adjusted phase derivative value set (60) using a stretching factor (70) and obtaining an output signal representation value set (80),
F. Synthesizing using said output signal representation value set (80) and obtaining an audio output signal (90) with a corrected phase.

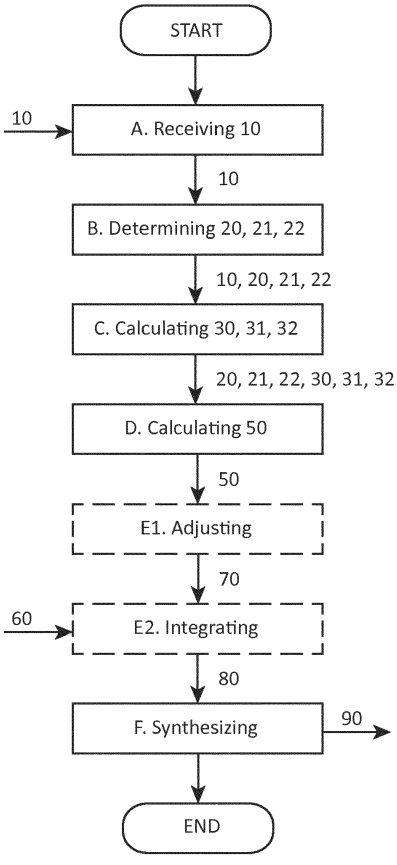



*Fig. 2*

EP 3 447 767 A1

## Description

### Field of the invention and description of prior art

**[0001]** The present invention relates to a method for phase correction in a phase vocoder.

**[0002]** Furthermore, the invention relates to a device for carrying out a method according to the invention.

**[0003]** The phase modification with a phase vocoder (PV) is a widely spread technique for processing audio signals. It employs a short-time Fourier transform (STFT) analysis-modify-synthesis loop and is typically used for time-scaling of signals by means of using different time steps for STFT analysis and synthesis. The main challenge of PV used for that purpose is the correction of the STFT phase.

**[0004]** The term phase vocoder was coined by Flanagan and Golden [1] but the now classical form of PV employing STFT analysis and synthesis using different analysis and synthesis steps for the purposes of time-scaling was introduced later by Portnoff [2, Section 6.4.4]. The way how the phase is corrected in the classical PV is based on the linear phase progression of a sinusoid with constant frequency. Each frequency channel is treated as a separate sinusoid whose phase is computed by accumulating the estimate of its instantaneous frequency and thus preserving the horizontal phase coherence [3]. Obviously, such an approach cannot cope well with time-varying sinusoidal signals and non-sinusoidal signals. A modification of PV was proposed by Laroche and Dolson [4, 5]. This improved PV involves spectral peak picking, tracking and "locking" the phase of the frequency bins belonging to a sinusoid to the phase of the peak (scaled phase locking). The phase locking enforces vertical phase coherence within the region of influence of the peak. Although the phase-locked PV is able to handle signals consisting of sinusoids with time-varying frequencies, it still fails for percussive sounds and transient events in general. A review of the phase vocoder, its history and alternative approaches to time and frequency scale modifications of audio signals can be found in [6, 7, 8].

**[0005]** Time scaling using phase vocoder techniques is known to produce specific artifacts such as transient smearing, "echo" and "loss of presence" collectively referred to as phasiness [4].

**[0006]** The artifacts are generally attributed to the loss of vertical phase coherence. Transient smearing compensation has been addressed by several authors. The most common approach is performing a phase reset or phase locking at transients [9,10,11]. Other approaches involve disabling the time-scaling at transients [12], or rely on using different window lengths for harmonic and transient parts [13,14]. All approaches mentioned rely on correct transient detection. The preservation of vertical phase coherence is considered to be important for the quality of time-scaled voiced speech signals. It has been reported that not preserving the relative phase shift between the fundamental and the partials is perceived as unnatural. Several specialized methods for time-stretching of monophonic voice signals were proposed [15,16]. Historically, time scaling techniques which preserve relative phase shift between the partials are called shape preserving [17]. To avoid dealing with phase altogether, a magnitude-only reconstruction has been considered [18]. Although efficient and real-time algorithms have recently been proposed, e.g. [19, 20], they do not perform favorably for large stretching factors. This is due to the magnitude not complying with the new synthesis step.

**[0007]** Prior art PV algorithms do not automatically enforce horizontal and vertical phase coherence for broadband, non-sinusoidal components. Moreover, explicit peak picking or transient detection is required.

### Summary of the invention

**[0008]** It is an object of the invention to overcome the disadvantages of the prior art.

**[0009]** In a first aspect of the invention, this aim is achieved by means of above-mentioned method, comprising the following steps:

A. Receiving an audio input signal,

B. Determining at least a first, a second and a third time frame of said audio input signal and obtaining a first, a second and a third time frame index parameter,

C. Calculating a first, a second and a third signal representation value set for said audio input signal related to each of said first, said second and said third time frame index parameter, wherein said first, said second and said third time frame of said audio input signal are matched to a set of frequency channels respectively, and the calculation of each of said first, said second and said third signal representation value set comprises signal representation parameters for each frequency channel of said set of frequency channels, derived from said audio input signal using a windowing function and a short-time Fourier transform,

D. Calculating a phase derivative value set by phase differentiation for each frequency channel of said set of frequency channels based on said first, said second and said third signal representation value set using finite differences,

E. Integrating said phase derivative value set and obtaining an output signal representation phase value set,

F. Synthesizing using said output signal representation phase value set and obtaining an audio output signal with a corrected phase.

**[0010]** The representation phase value set, said first, said second and said third signal representation value set as well as said output signal representation phase value set is a set of data, respectively, obtained by calculations from said audio input signal as explained in the subsequent description, which serves as a value set for the subsequent signal synthesis. Each of said representation phase value sets can comprise phase and magnitude values, which can be considered individually or in combination in the synthesis.

**[0011]** Hence, the invention provides a method for unsupervised time-stretching and pitch-shifting of audio signals.

**[0012]** The pitch-shifting is achieved after said step F of synthesizing by resampling the time-scaled audio input signal with respect to the original duration of the audio input signal. The pitch-shifting can be performed by oversampling or subsampling of the time-scaled audio input signal. The resampling ratio is tied up with the scaling factor. For example, in order to achieve a pitch-shifting by one octave up, a two times stretched signal must be subsampled by a factor of two.

**[0013]** The resampling can be also performed after said step A, which can be more efficient for stretching factors higher than one, i.e. a pitch-shifting up.

**[0014]** During resampling, new signal sample values for the further processing of the audio output signal are obtained from the input samples, for example by a polynomial interpolation. In a further development, especially for pitch-shifting up, an anti-aliasing filter could be employed.

**[0015]** For example, unmodified audio signals and time-stretched and/or pitch-shifted audio signals, modified according to the invention, can be concatenated to constitute a common audio signal.

**[0016]** Said step E of integrating said phase derivative value set can be further developed into two steps, namely:

E1. Adjusting said phase derivative value set and obtaining an adjusted phase derivative value set,

E2. Integrating said adjusted phase derivative value set using a stretching factor and obtaining a representation phase value set.

**[0017]** Moreover, this aim of the invention is achieved by a device comprising a memory for storing a first, a second and a third signal representation value set, and a program logic for carrying out the method according to the invention, and a processor for carrying out the program logic and/or the method according to the invention.

**[0018]** In order to obtain a longer output signal, said stretching factor or the time scaling factor is greater than one, and for a shorter output signal said stretching factor or the time scaling factor is lower than one.

**[0019]** Further, it is beneficial, when said step E of integrating said phase derivative value set is performed through a phase gradient heap integration algorithm, preferably through a real-time phase gradient heap integration algorithm. In other words, it is advantageous, when the PGHI algorithm [21], originally proposed for magnitude-only reconstruction, is a real-time phase gradient heap integration algorithm (RTPGHI) [19]. The same applies when using said steps E1 and E2 using said adjusted phase derivative value set.

**[0020]** In an enhancement of the method according to the invention, said method further including, subsequent to said step F of synthesizing, the step of:

G. Resampling of said audio output signal, wherein

said first, said second and/or said third time frame index parameter constitute a first time-base of said audio input signal, and

said stretching factor constitutes a second time-base of said audio output signal, and

said audio output signal is resampled using said first time-base and obtaining a resampled audio signal with pitch-shifted content.

**[0021]** Further, it is beneficial, if at said step G of resampling, a polynomial interpolation is used to obtain said resampled audio signal, and preferably, further an anti-aliasing filter is used to obtain said resampled audio signal.

**[0022]** It is advantageous, when the phase derivative comprises a time domain partial derivative and a frequency domain partial derivative for each frequency channel of said set of frequency channels, respectively. Hence, superior audio results in the phase corrected audio signal are obtained.

**[0023]** For computing said time domain phase derivative using the centered differences, it is mandatory to use three

time frames, i.e. both of the equations (13) and (14) are used in equation (15) and frames with time indices $n$ - 1, $n$, and $n$ + 1 are needed.

**[0024]** At said step D of calculating when using finite differences, it is preferred using centered finite differences. Alternatively, any method according to (13), (14) and (15) can be used.

**[0025]** Considering both phase partial derivatives is necessary for achieving high quality for non-sinusoidal signals, for which said frequency direction derivative is non-zero. Algorithms PGHI or RTPGHI were designed to consider both phase derivatives and they switch between them according to the signal value set frequency energy distribution.

**[0026]** It is even more beneficial for a simple implementation of the algorithm, when said time domain partial derivative is obtained by using said first, said second and said third signal representation value set.

**[0027]** It is even more beneficial for a simple implementation of the algorithm, when said frequency domain partial derivative is obtained by using frequency channels adjacent to each other of said set of frequency channels from said second signal representation value set with the same time frame index parameter.

**[0028]** It is beneficial, when said number of frequency channels of said set of frequency channels is set to a value between 512 channels and 16368 channels, preferably between 4096 channels and 16368 channels.

**[0029]** It is beneficial, at said step B at determining a first, a second and a third signal representation value set from said audio input signal, when said windowing function covers a time range of 40 ms up to 160 ms of said audio input signal, preferably 90 ms.

**[0030]** It is beneficial, at said step B at determining a first, a second and a third signal representation value set from said audio input signal, when said windowing function is a Hann window, preferably with a sample size set to a value between 512 samples and 16368 samples, more preferably between 2048 samples and 8192 samples.

**[0031]** It is beneficial, when said stretching factor is set to a value between 10 and 1/5.

**[0032]** It is beneficial, at said step F of synthesizing, when a synthesis time step is set to a value between 256 and 16368, preferably between 512 and 2048. The sampling rate is fixed at said step B of said method, where the audio input signal is sampled e.g. by a sampling unit and converted into said signal representation value sets. Said sampling rate connects the time frame index parameters with the chronological properties of said audio input signal.

**[0033]** It is beneficial, when the received audio input signal is non-stationary, preferably with non-periodic characteristics.

**[0034]** It is beneficial, when said number of frequency channels is equal or higher than the number of samples of the window of said windowing function.

**[0035]** It is beneficial, when the synthesis time step is ¼ of the window length of said windowing function. Hence, a constant overlap-add property of the Hann window can be achieved.

**[0036]** The inventors have observed that using more frequency channels than it is necessary, e.g. when PGHI or RTPGHI in a time-stretching or a pitch-shifting method, respectively, are used, twice as many than the samples of the window of said windowing function, reduces the occurrence of artifacts .

**[0037]** The method for phase correction in the PV, relies on both partial derivatives of the STFT phase and their integration. The method is efficient in the sense that it works automatically and it does not require analyzing the signal content.

## Brief description of the drawings

**[0038]** The specific features and advantages of the present invention will be better understood through the following description. In the following, the present invention is described in more detail with reference to exemplary embodiments (which are not to be construed as a limitation) depicted in the drawings, which show in:

Fig. 1 a block diagram of the arrangement for executing the method;

Fig. 2 a flow chart of the method according to the invention;

Fig. 3 a spectrogram of a sum of a pure sinusoid, an exponential chirp and an impulse; the values are in dB;

Fig. 4 a scaled time phase derivative $\frac{f_s}{2\pi}\left(\Delta_f \Phi_a\right)$ (instantaneous frequency); $f_s$ stands for the sampling rate and the values are in Hertz;

Fig. 5 a scaled absolute value of a frequency phase derivative $\frac{10^3 L}{2\pi f_s}\left|\Delta_f \Phi_a\right.$ (absolute value of the local group delay); note that $(\Delta_f \Phi_a)$ is zero for the pure sinusoid and the values are in milliseconds;

Fig. 6 Conceptual spectrograms overlaid with the phase spreading paths:

a) pure sinusoid;
b) linear chirp;
c) two sinusoids;
d) before impulse peak;
e) after impulse peak;

Fig. 7 Conceptual phase of STFT frames with identified equations used to compute phase derivatives:

a) computation of a time direction/ domain derivative;
b) computation of a frequency direction/ domain derivative;

Fig. 8 a flow chart of an algorithm for practicing the method according to the invention.

Detailed description of the invention

**[0039]** Fig. 1 shows a block diagram of a device of a phase vocoder for executing the method according to the invention, comprising a vocoder device 100 with a memory 110. Said device is configured to receive an audio input signal 10, to process said audio input signal 10 and to obtain an audio output signal 90. Said method can be executed as long as the input signal 10 is received, which is typically a non-stationary signal, and preferably with non-periodic characteristics.

**[0040]** The device 100 comprises a memory 110 for storing a first, a second and a third signal representation value set 30, 31, 32, and a program logic 120 for carrying out the method according to the invention, and a processor 130 for carrying out the program logic 120 and/or the method according to the invention.

**[0041]** Fig. 2 shows a flow chart of the method according to the invention, which is based on the following principle:

**[0042]** Given a discrete time signal $f$ which is nonzero on the interval 0... $L$ - 1, a real-valued analysis window $g_a$ concentrated around the origin and an analysis time step $a_a$, the discrete STFT is given by

$$c(m,n) = \sum_{l \in Z} f(l + na_a) g_a(l) e^{\frac{-i2\pi ml}{M}} \quad (1)$$

**[0043]** For $m$ = 0... $M$ - 1, M being the FFT length, and $n$ = 0... $N$ - 1, where $N = \frac{L}{a_a}$ is the number of STFT frames. Setting $M = L$ results in the full frequency resolution, but, typically, it is chosen such that $M << L$. The analysis frequency step is defined as

$$b_a = \frac{L}{M} \quad (2)$$

**[0044]** The magnitude s and phase $\Phi_a$ components of the STFT can be separated by

$$s(m,n) = |c(m,n)| \quad (3)$$

$$\Phi_a(m,n) = arg\big(c(m,n)\big) \quad (4)$$

**[0045]** A time scaling factor is defined as

$$\alpha = \frac{a_s}{a_a} \quad (5)$$

where $a_s$ denotes the synthesis time step. The length of the output signal is therefore equal to $\alpha L$ and the synthesis frequency step to

$$b_s = \alpha b_a \tag{6}$$

**[0046]** A synthesis phase is constructed by the recursive *phase propagation* (integration) formula [5].

$$\Phi_s(m,n) = \Phi_s(m,n-1) + a_s(\Delta_t \Phi_a)(m,n) \tag{7}$$

where $\Delta_t$ performs differentiation of the analysis phase $\Phi_a$, or alternatively, by employing the trapezoidal integration rule, by

$$\Phi_s(m,n) = \Phi_s(m,n-1) + \frac{a_s}{2}\left((\Delta_t \Phi_a)(m,n-1) + (\Delta_t \Phi_a)(m,n)\right) \tag{8}$$

**[0047]** A time-scaled signal $\tilde{f}$ is reconstructed using

$$\tilde{f}(l) = \sum_{n=0}^{N-1} \tilde{f}_n(l - na_s) \tag{9}$$

for $l$ = 0 ... $\alpha L$ - 1 with

$$\tilde{f}_n(l) = g_s(l) \sum_{m=0}^{M-1} s(m,n) e^{-i\Phi_a(m,n)} e^{\frac{-i2\pi ml}{M}} \tag{10}$$

and with

$$g_s(l) = \frac{1}{M} \frac{g_a(l)}{\sum_{n\in Z} g_a(l-na_s)^2} \tag{11}$$

being the synthesis window $g_s$. To summarize, the classical PV performs differentiation and back integration of the phase in the time direction for all frequency bins.

The formula (10) can be interpreted as a sampling of the continuous STFT which is by itself a two-dimensional function of time and frequency. Therefore, $(\Delta_t \Phi_a)$ can be interpreted as an approximation of the partial derivative of the phase in time. The complete phase gradient however involves also the partial derivative in frequency whose approximation will further be denoted as $(\Delta_f \Phi_a)$. This second gradient component is completely disregarded in the classical PV, which introduces significant inaccuracies except for pure, stationary sinusoids. For the latter, $(\Delta_f \Phi_a)$ indeed equals zero.

**[0048]** It is noted that said steps of the method according to Fig. 2 can be repeated as long as said audio input signal 10 feeds at least three time frames into the process.

**[0049]** The example in Fig. 3 shows plots of a spectrogram (magnitude of coefficients of STFT) of a sum of a pure sinusoid, an exponential chirp and an impulse and Fig. 4 and 5 show the partial derivatives of the phase. In particular, Fig. 5 shows that, for the chirp and pulse components, $(\Delta_f \Phi_a)$ unsurprisingly has non-negligible values.

**[0050]** However, the formulas for estimating partial derivatives of the STFT phase $(\Delta_f \Phi_a)$ and $(\Delta_t \Phi_a)$ can be derived as well as an adaptive integration algorithm, which takes them both into account to produce an adjusted synthesis phase $\Phi_s$ to be used in the formula (10), as can be seen subsequently.

**[0051]** The formulas exploit the well-known phase *unwrapping* procedure which involves taking the principal argument of an angle. The notation from [22] is adopted and the principal argument is defined as

$$[x]_{2\pi} = x - 2\pi\left[\frac{x}{2\pi}\right] \tag{12}$$

where $[\cdot]$ denotes rounding towards the closest integer. The well-known phase differentiation in the time direction (involving conversion to *heterodyned* [5] phase difference and back) can be written as

$$\left(\Delta_{t,back} \Phi_a\right)(m,n) = \frac{1}{a_a}\left[\Phi_a(m,n) - \Phi_a(m,n-1) - \frac{2\pi m a_a}{M}\right]_{2\pi} + \frac{2\pi m}{M} \tag{13}$$

which is a modified backward difference scheme. The forward difference scheme can be written similarly

$$\left(\Delta_{t,fwd}\Phi_a\right)(m,n) = \frac{1}{a_a}\left[\Phi_a(m,n+1) - \Phi_a(m,n) - \frac{2\pi m a_a}{M}\right]_{2\pi} + \frac{2\pi m}{M} \tag{14}$$

and the centered difference is simply an average of the two

$$\left(\Delta_{t,cent}\Phi_a\right)(m,n) = \frac{1}{2}\left(\left(\Delta_{t,back}\Phi_a\right)(m,n) + \left(\Delta_{t,fwd}\Phi_a\right)(m,n)\right) \tag{15}$$

**[0052]** Any of the schemes can be used in place of $\Delta_t$. In our experience $\Delta_{t,cent}$ is the most suitable. Similarly, the differentiation in the frequency direction can be performed using the backward scheme

$$\left(\Delta_{f,back}\Phi_a\right)(m,n) = \frac{1}{b_a}\left[\Phi_a(m,n) - \Phi_a(m-1,n)\right]_{2\pi} \tag{16}$$

the forward scheme

$$\left(\Delta_{f,fwd}\Phi_a\right)(m,n) = \frac{1}{b_a}\left[\Phi_a(m+1,n) - \Phi_a(m,n)\right]_{2\pi} \tag{17}$$

or the centered scheme

$$\left(\Delta_{f,cent}\Phi_a\right)(m,n) = \frac{1}{2}\left(\left(\Delta_{f,back}\Phi_a\right)(m,n) + \left(\Delta_{f,fwd}\Phi_a\right)(m,n)\right) \tag{18}$$

**[0053]** Having the means for estimating the phase partial derivatives, the RTPGHI algorithm [19] can be invoked with few modifications. In its essence, the method proceeds by processing one frame at a time computing the synthesis phase of all frequency channels of the current $n$-th frame $\Phi_s(\cdot,n)$. It requires storing the already computed phase $\Phi_s(\cdot,n-1)$ and the time derivative $(\Delta_t\Phi_a)(\cdot,n-1)$ of the previous $(n-1)$-th frame and further, it requires access to the coefficients of the previous, current and one "future" frame ($c(\cdot,n-1)$, $c(\cdot,n)$ and $c(\cdot,n+1)$) assuming the centered differentiation scheme (15) is used for computing $(\Delta_t\Phi_a)(\cdot,n)$. Before the algorithm starts $(\Delta_t\Phi_a)(m,n)$ and $(\Delta_t\Phi_a)(m,n)$ are computed for all $m$ and current $n$. The algorithm starts by selecting the frequency bin $m_h$ of the coefficient with the highest magnitude from the previous $(n-1)$-th frame and propagates the phase along the time direction to the current frame using (8) such that $\Phi_s(m_h, n)$ is obtained. Up to this point, the procedure is equivalent with the classical PV. The way how the phase of the remaining coefficients is obtained is different since the phase can now be propagated also in the frequency direction from the already computed phase in the current frame. The phase propagation direction is decided on-the-fly according to the magnitude of the coefficients.

**[0054]** Based on the aforesaid principle, the method according to the invention can be explained by following steps according to Fig. 2 as one embodiment of the invention:

A. Receiving an audio input signal 10,

B. Determining at least a first, a second and a third time frame of said audio input signal 10 and obtaining a first, a second and a third time frame index parameter 20, 21, 22,

C. Calculating a first, a second and a third signal representation value set 30, 31, 32 for said audio input signal 10 related to each of said first, said second and said third time frame index parameter 20, 21, 22, wherein said first, said second and said third time frame of said audio input signal 10 are matched to a set of frequency channels 40, 41, 42 respectively, and the calculation of each of said first, said second and said third signal representation value set 30, 31, 32 comprises signal representation parameters for each frequency channel 0-6 out of a set of frequency channels 40, 41, 42, derived from said audio input signal 10 using a windowing function and a short-time Fourier transformation,

D. Calculating a phase derivative value set 50 by phase derivatives for each frequency channel 0-6 of said set of frequency channels 40, 41, 42 based on said first, said second and said third signal representation value set 30, 31, 32 using centered finite differences,

E. Integrating said phase derivative value set 50 using a stretching factor 70 and obtaining an output signal representation value set 80,

F. Synthesizing using said output signal representation value set 80 and obtaining an audio output signal 90 with a corrected phase.

**[0055]** It is beneficial using the shown sequence of said steps of the method accordingly.

**[0056]** Said first, said second and said third signal representation value set 30, 31, 32 can be stored in a memory 110 of a vocoder device 100.

**[0057]** It is beneficial, if in said step B obtained said second and said third time frame index parameter 21, 22 related to the $n$-th and the ($n$ + 1)-th frame, as well as in said step C calculated said second and said third signal representation value set 31, 32 related to the $n$-th and the ($n$ + 1)-th frame can be stored in said memory 110 and can be loaded later on from said memory 110 by said processor 130 at the next succeeding pass of said method with said steps A to F and said second and said third time frame index parameter 21, 22 and said second and said third signal representation value set 31, 32 serve as respective values related to the ($n$ - 1)-th and the $n$-th frame.

**[0058]** The same applies to said step D, wherein said calculated phase derivative value set 50 related to the actual $n$-th and the ($n$ + 1)-th frame can be stored in said memory 110 and can be loaded later on from said memory 110 by said processor 130 at the next succeeding pass of said method with said steps A to F and said phase derivative value set 50 serve as respective values related to the ($n$ - 1)-th and the $n$-th frame.

**[0059]** Thus, by storing and reloading already calculated values of the phase derivative set and/ or signal representation value sets, the processing efficiency when carrying out said method can be improved.

**[0060]** It is beneficial, if said first, said second and said third frame index parameters 20, 21, 22 correspond to consecutive time frames such that said first frame index parameter 20 is the oldest and said third frame index parameters 22 is the most recent one.

**[0061]** Said step E of integrating said phase derivative value set 50 can be further developed into and carried out by two steps, namely:

E1. Adjusting said phase derivative value set 50 and obtaining an adjusted phase derivative value set 60,

E2. Integrating said adjusted phase derivative value set 60 using a stretching factor 70 and obtaining an output signal representation value set 80.

**[0062]** It is beneficial using the shown sequence of said steps of the method accordingly.

**[0063]** Said step C of calculating a signal representation value set 30, 31, 32 is a short-time Fourier transform, which can be performed by following steps:

C1. Divide an input signal 10 into consecutive overlapping blocks (frames) whose length is equal to the analysis window length and whose distances are $a_a$ samples (may vary over time),

C2. Weight each block with an analysis window,

C3. Zero-pad each weighted block such that the length is equal to the required number of frequency channels 0-6,

C4. Circularly shift the extended block such that the position of the peak of the window is moved to the beginning of the extended block,

C5. Apply the fast Fourier transform to the weighted, extended and circularly shifted array,

C6. Separate magnitude and phase of the complex Fourier coefficients.

**[0064]** It is beneficial using the shown sequence of said steps of the method accordingly.

**[0065]** Said step D of calculating a phase derivative value set 50 can be performed by following steps:

D1. Time direction differences (i.e. between frames) of phase according to (13) or (14) or (15),

D2. Frequency direction differences (i.e. between frequency bins) of phase according to (16) or (17) or (18).

**[0066]** It is beneficial using the shown sequence of said steps of the method accordingly.

**[0067]** The optional steps E1 and E2 of adjusting said phase derivative value set 50 can be performed by adjusting the phase derivative and the magnitude of the parameters of said phase derivative value set 50. Said stretching factor 70 is an input parameter, which can be set e.g. by a user of the method and is used to stretch said audio input signal as requested. Said stretching factor 70 can compress or expand the time base of said audio input signal.

**[0068]** Adjusting said phase derivative value set 50 can be also distributed to said step D of calculating a phase derivative value set 50 and said step E2 of integrating said adjusted phase derivative value set 60, and it can be just a scalar multiplication.

**[0069]** Said step E2 of integrating said phase derivative value set 50 said adjusted phase derivative value set 60 of the current frame can be performed by following steps:

E.a., E2.a. Access to the new representation phase of the previous frame,

E.b., E2.b. Determine the magnitude of the representation of the current and the previous frames,

E.c., E2.c. Determine representation's phase derivatives for the current and the previous frames,

E.d., E2.d. Run Algorithm 1.

**[0070]** It is beneficial using the shown sequence of said steps of the method accordingly.

**[0071]** If the length of the signal is unknown at the time of processing of the current frame, the scaling by $\frac{1}{b_\alpha}$ in (16) or (17) and by $b_s$ on lines 18 and 23 in Algorithm 1 can be consolidated into the multiplication by the scaling factor $\alpha = \frac{b_s}{b_\alpha}$, which is denoted also with the numeral 70 and which is independent of the signal length.

**[0072]** Said step F of signal synthesis (inverse short-time Fourier transform) can be performed by following steps:

F1. For each frame, combine the magnitude with the new representation phase,

F2. Apply inverse fast Fourier transform,

F3. Undo circular shift from said step of calculating said second signal representation value set 31 as said step C4,

F4. Weight the initial part of the signal with the synthesis window and crop out the remaining part.

F5. Overlap-add the resulting array to the output signal.

**[0073]** At said step F3, said second signal representation value set 31 corresponds to the *n*-th frame, which is synthesized.

**[0074]** It is beneficial using the shown sequence of said steps of the method accordingly.

**[0075]** Note that the algorithm employs trapezoidal integration rule for the phase propagation on lines 11,18 and 23. Line 11 implements (8), but the integration is also performed in the vertical (frequency) direction (lines 18, 23) employing the frequency phase derivative estimate ($\Delta_f \Phi_a$). The phase for each frequency bin is however computed only once. Which equation is used is decided adaptively according to the magnitude s with the help of the *max heap* [23]. The max heap is a dynamic data structure that always places coordinates (*m, n*) of the coefficient with the highest magnitude on top. It is populated by the coordinates of the coefficients with already known phase which are possible candidates from which the phase can be propagated further. At the beginning of the algorithm, all frequency bins from the previous frame with significant magnitude are inserted into the heap (line 5). In the first iteration, the only possible phase propagation direction is the time direction on line 11. The coordinates of the just computed coefficient are inserted into the heap (line 13). In the next iteration, two options are possible. Either the same procedure is repeated with another frequency bin from the previous frame, or the extracted top of the heap belongs to the current frame n. In that case, the phase is propagated in the frequency direction to both neighboring frequency bins on lines 18 and 23. Both neighbors are then inserted into the heap and in the next step, the integration continues with extracting the new heap top and spreading the phase further.

**[0076]** Fig. 6 a) to e) show examples of how the phase is spread for several signals with frames *n* - 1 and *n*. The orientation of the arrow corresponds to the line in the algorithm as follows: 11 (horizontally, →), 18 (vertically upwards, ↑) and 23 (vertically downwards, ↓).

**[0077]** Each row of each example in Fig. 6 a) to e) represents one frequency channel 0-6, which equals the integer index *m* in the previous formulas. Further, said first and second time frame index factor 20, 21 equals the integer index *n* - 1 and *n* in the previous formulas, respectively.

**[0078]** Each column of each example Fig. 6 a) to e) represents one set of frequency channels 40, 41 at different points of time, i.e. time frame index parameters 20, 21 respectively.

**[0079]** Said time domain partial derivative 35 and said frequency domain partial derivative 36 is indicated in Fig. 6 a) to e). Within a practical implementation, said time and frequency domain phase derivatives 35, 36 are arrays. It is noted, that said time domain phase derivatives 35 correspond to said horizontally time direction components and said frequency domain phase derivatives 36 correspond to said vertically (upwards and downwards) frequency direction components as explained above.

**[0080]** Fig. 7 a) and b) conceptually depicts an example of the phase of STFT frames *n* - 1, *n* and *n* + 1according to equation (4). The arrows with dashed lines in Fig. 7 a) identify the equations and said frame indices used in the computation of said time domain phase derivatives. Similarly, the arrows with dashed lines in Fig. 7 b) identify the equations and said frequency channels used for computing said frequency domain phase derivatives. For simplicity, only the channel with index *n* is depicted. In both cases, the derivatives are computed for all *m* values.

**[0081]** Each column of the example Fig. 7 a) represents one set of frequency channels 40, 41, 42 at different points of time, i.e. time frame index parameters 20, 21, 22, respectively, which equal the integer index *n* - 1, *n*, and *n* + 1 in the previous formulas, respectively.

**[0082]** Similarly, only the *n*-th frame is required for computing said frequency domain derivative.

**[0083]** Summarized, three frames *n* - 1, *n* and *n* + 1 are required for computing a new phase of the *n*-th frame, but only the *n*-th frame is synthesized at a point, i.e. one pass of said method. Consequently, the length of said audio output signal 90 is shorter than the length of said audio input signal 10. Moreover, said audio output signal 90 becomes zero if said audio input signal 10 is shorter than a three frames length.

**[0084]** In Fig. 8 said step E2 of integrating said adjusted phase derivative value set 60 and obtaining an output signal representation value set 80 is shown in detail by a flowchart representing Algorithm 1, which shows the phase gradient heap integration for *n*-th frame.

**Input:** Phase time derivative $(\Delta_t\phi_a)$ and magnitude $s$ of frames $n$ and $n-1$, phase frequency derivative $(\Delta_f\phi_a)$ for frame $n$, estimated phase $\phi_s$ for frame $n-1$ and relative tolerance *tol*.

**Output:** Phase estimate $\phi_s$ for frame $n$.

$abstol \leftarrow tol \cdot \max(s(m,n) \cup s(m,n-1))$;
Create set $\mathcal{I} = \{m : s(m,n) > abstol\}$;
Assign random values to $\phi_s(m,n)$ for $m \notin \mathcal{I}$;
Construct a self-sorting max *heap* [23] for $(m,n)$ tuples;
Insert $(m,n-1)$ for $m \in \mathcal{I}$ into the *heap*;
**while** $\mathcal{I}$ *is not* $\emptyset$ **do**
  **while** heap *is not empty* **do**
    $(m_h, n_h) \leftarrow$ remove the top of the *heap*;
    **if** $n_h = n-1$ **then**
      **if** $(m_h, n) \in \mathcal{I}$ **then**
        $\phi_s(m_h,n) \leftarrow \phi_s(m_h,n-1) + \frac{a_s}{2}((\Delta_t\phi_a)(m_h,n-1) + (\Delta_t\phi_a)(m_h,n))$;
        Remove $(m_h,n)$ from $\mathcal{I}$;
        Insert $(m_h,n)$ into the *heap*;
      **end**
    **end**
    **if** $n_h = n$ **then**
      **if** $(m_h+1, n) \in \mathcal{I}$ **then**
        $\phi_s(m_h+1,n) \leftarrow \phi_s(m_h,n) + \frac{b_s}{2}((\Delta_f\phi_a)(m_h,n) + (\Delta_f\phi_a)(m_h+1,n))$;
        Remove $(m_h+1,n)$ from $\mathcal{I}$;
        Insert $(m_h+1,n)$ into the *heap*;
      **end**
      **if** $(m_h-1, n) \in \mathcal{I}$ **then**
        $\phi_s(m_h-1,n) \leftarrow \phi_s(m_h,n) - \frac{b_s}{2}((\Delta_f\phi_a)(m_h,n) + (\Delta_f\phi_a)(m_h-1,n))$;
        Remove $(m_h-1,n)$ from $\mathcal{I}$;
        Insert $(m_h-1,n)$ into the *heap*;
      **end**
    **end**
  **end**
**end**

Algorithm 1: Phase gradient heap integration for *n*-th frame

**[0085]** In the shown embodiment of the invention, adjusting of said phase derivative value set 50 and integrating said adjusted phase derivative value set 60 is incorporated in the method and said related featuring steps E1 and E2 are marked with dashed lines in Fig. 8.

**[0086]** The input parameters of Algorithm 1 are the representation phase of the previous frame $\Phi_s(\cdot, n-1)$, the magnitude of the representation of the current and the previous frames, the representation's phase derivatives for the current and the previous frames and the relative magnitude tolerance *tol.* Said algorithm delivers the phase of the current frame $\Phi_s(\cdot, n)$.

**[0087]** Algorithm 1 can be illustrated by following steps:

E.d.1. Setup:

- Find the maximum magnitude $s_{max}$ of coefficients from the previous and current frames,
- $abstol = tol \cdot s_{max}$,
- Identify a set of coefficients from the current frame with magnitude above *abstol,*
- Assign random values to coefficients below *abstol.*

E.d.2. Build heap:

- Construct a max heap,
- Populate it with indices of the coefficients from the previous frame for which their direct neighbors in the current frame are above *abstol.*

E.d.3. Check whether remaining unknown phase values exist?

- If not, terminate the algorithm.

E.d.4. Max heap:

- Extract and remove the maximum from the heap.

E.d.5. Check whether the maximum is in the previous frame?

- If so:

  E.d.5a. Check whether the phase in the horizontal direction is already computed?

  - If not, propagate phase in the horizontal direction (integration using the time-direction phase derivative) and insert an index of the just computed coefficient into the heap,
  - Else skip the last instance E.d.5a (propagate phase in the horizontal direction is already computed).

- Else

  E.d.5b. Check whether the phase in the vertical direction upwards is already computed?

  - If not, propagate phase in the vertical direction upwards (integration using the time-direction phase derivative) and insert an index of the just computed coefficient into the heap,
  - Else skip the last instance E.d.5b (propagate phase in the vertical direction upwards is already computed).

  E.d.5c. Check whether the phase in the vertical direction downwards is already computed?

  - If not, propagate phase in the vertical direction downwards (integration using the time-direction phase derivative) and insert index of the just computed coefficient into the heap,
  - Else skip the last instance E2.d.5c (propagate phase in the vertical direction downwards is already computed).

**[0088]** Said steps E.d.1-E.d.5 of Algorithm 1 related to said step E can be implemented accordingly as steps E2.d.1-E2.d.5 related to said step E2.

**[0089]** At a glance, said step E of integrating said phase derivative value set 50 is performed through a phase gradient heap integration algorithm, preferably through a real-time phase gradient heap integration algorithm.

**[0090]** Further, the phase derivative of said first, said second or said third signal representation value set 30, 31, 32 comprises a time domain partial derivative 35 and a frequency domain partial derivative 36 for each frequency channel 0-6 of said set of frequency channels 40, 41, 42, respectively.

**[0091]** Moreover, said time domain partial derivative 35 is obtained by using said first, said second and said third signal representation value set 30, 31, 32 of instantaneously succeeding time frame index parameters 20, 21 and 22, i.e. $n$ - 1, $n$, and $n$ + 1.

**[0092]** Further, said frequency domain partial derivative 36 is obtained by using frequency channels 0-6 adjacent to each other of said set of frequency channels 41 from said second signal representation value set 31, with the same time frame index parameter 21, which corresponds to the $n$-th frame.

**[0093]** The inventors have observed that it is beneficial, if aforesaid preferred calculation parameters are combined, wherein at said step B of determining said signal representation value sets 30, 31, 32 from said audio input signal 10 is based on a 4092 samples long Hann window, said number of said frequency channels of said set of frequency channels 40, 41, 42 is set to M = 8192 and at said step F of synthesizing, the synthesis time step is fixed to $a_s$ = 1024 and $a_a$ is changed according to (5) rounded to the closest integer. Preferably said audio input signal 10 is sampled at 44.1 kHz. The tolerance in Algorithm 1 was set to $tol = 10^{-6}$. Said stretching factor 70 is preferably set to a value between 10 (ten times longer) and 1/5 (five times shorter).

**[0094]** Further, in a not shown embodiment, said method according to the invention further including, subsequent to said step F of synthesizing, the step of:

G. Resampling of said audio output signal 90, wherein

said first, said second and/or said third time frame index parameter 20, 21, 22 constitute a first time-base of said audio input signal 10, and said stretching factor 70 constitutes a second time-base of said audio output signal 90, and

said audio output signal 90 is resampled using said first time-base and obtaining a resampled audio signal with pitch-shifted content.

**[0095]** At said step G of resampling, a polynomial interpolation and an anti-aliasing filter is used to obtain said resampled audio signal. For pitch-shifting up, the signal is stretched by a stretching factor greater than one and subsampled by the same ratio. For pitch-shifting down, the signal is compressed with a stretching factor lower than one and oversampled by the same ratio.

**[0096]** For pitch-shifting up, the cutoff frequency of the low-pass anti-aliasing filter should be close to the sampling rate divided by two times the resampling ratio.

**References:**

**[0097]**

[1] J. L. Flanagan and R. M. Golden, "Phase vocoder", Bell System Technical Journal, vol. 45, no. 9, pp. 1493-1509,1966.

[2] M. R. Portnoff, "Time-scale modification of speech based on short- time Fourier analysis", Ph.D. dissertation, Massachusetts Institute of Technology, 1978.

[3] E. Moulines and J. Laroche, "Non-parametric techniques for pitch-scale and time-scale modification of speech", Speech Communication, vol. 16, no. 2, pp. 175 - 205,1995.

[4] J. Laroche and M. Dolson, "Phase-vocoder: About this phasiness busi- ness", in IEEE ASSP Workshop on Applications of Signal Processing to Audio and Acoustics, Oct 1997, p. 4.

[5] ---, "Improved phase vocoder time-scale modification of audio", IEEE Transactions on Speech and Audio Processing, vol. 7, no. 3, pp. 323 - 332, May 1999.

[6] U. Zölzer, Ed., DAFX: Digital Audio Effects. New York, NY, USA: John Wiley & Sons, Inc., 2002.

[7] M. Liuni and A. Röbel, "Phase vocoder and beyond", Musica/Tecnologia, vol. 7, no. 0, 2013.

[8] J. Driedger and M. Müller, "A review of time-scale modification of music signals", Applied Sciences, vol. 6, no. 2, 2016.

[9] C. Duxbury, M. Davies, and M. B. Sandler, "Improved time-scaling of musical audio using phase locking at transients", in Audio Engineering Society Convention 112, Apr 2002.

[10] A. Röbel, "A new approach to transient processing in the phase vocoder", in Proc. Int. Conf. Digital Audio Effects (DAFx-03), Sep 2003.

[11] E. Ravelli, M. Sandler, and J. P. Bello, "Fast implementation for non- linear time-scaling of stereo signals", in Proc. Int. Conf. Digital Audio Effects (DAFx-05), Sep 2005.

[12] F. Nagel and A. Walther, "A novel transient handling scheme for time stretching algorithms", in Audio Engineering Society Convention 127, Oct 2009.

[13] J. Driedger, M. Müller, and S. Ewert, "Improving time-scale modification of music signals using harmonic-percussive separation", IEEE Signal Processing Letters, vol. 21, no. 1, pp. 105-109, Jan 2014.

[14] E. S. Ottosen and M. Dörfler, "A phase vocoder based on nonstationary Gabor frames", CoRR, vol. abs/1612.05156, 2016. [Online]. Available: http://arxiv.org/abs/1612.05156

[15] A. Röbel, "A shape-invariant phase vocoder for speech transformation", in Proc. Int. Conf. Digital Audio Effects (DAFx-10), Graz, Austria, Sep. 2010, pp. 1-1.

[16] A. Moinet and T. Dutoit, "PVSOLA: A phase vocoder with synchronized overlap-add", in Proc. Int. Conf. Digital Audio Effects (DAFx-11), Sep. 2011.
[17] T. F. Quatieri and R. J. McAulay, "Shape invariant time-scale and pitch modification of speech", IEEE Transactions on Signal Processing, vol. 40, no. 3, pp. 497-510, March 1992.
[18] D. Griffin and J. Lim, "Signal estimation from modified short-time Fourier transform", IEEE Transactions on Acoustics, Speech and Signal Processing, vol. 32, no. 2, pp. 236-243, Apr 1984.
[19] Z. Průša and P. L. Søndergaard, "Real-time spectrogram inversion using phase gradient heap integration", in Proc. Int. Conf. Digital Audio Effects (DAFx-16), Sep 2016, pp. 17-21.
[20] Z. Průša and P. Rajmic, "Toward high-quality real-time signal reconstruction from STFT magnitude", IEEE Signal Processing Letters, vol. 24, no. 6, June 2017.
[21] Z. Průša, P. Balazs, and P. L. Søndergaard, "A Noniterative Method for Reconstruction of Phase from STFT Magnitude", IEEE/ACM Trans. on Audio, Speech, and Lang. Process., vol. 25, no. 5, May 2017.
[22] S. Kraft, M. Holters, A. von dem Knesebeck, and U. Zölzer, "Improved PVSOLA time-stretching and pitch-shifting for polyphonic audio", in Proc. Int. Conf. Digital Audio Effects (DAFx-12), Sep 2012.
[23] J. W. J. Williams, "Algorithm 232: Heapsort", Communications of the ACM, vol. 7, no. 6, pp. 347-348,1964.

**List of reference numerals:**

**[0098]**

| | |
|---|---|
| 0-6 | frequency channel |
| 10 | audio input signal |
| 20, 21, 22 | time frame index parameter |
| 30, 31, 32 | signal representation value set |
| 35 | time domain partial derivative |
| 36 | frequency domain partial derivative |
| 40, 41, 42 | set of frequency channels |
| 50 | phase derivative value set |
| 60 | adjusted phase derivative value set |
| 70 | stretching factor or time scaling factor |
| 80 | output signal representation value set |
| 90 | audio output signal |
| 100 | device, vocoder |
| 110 | memory |
| 120 | program logic |
| 130 | processor |

## Claims

**1.** A method for phase correction in a phase vocoder (100), said method comprising the steps of:

A. Receiving an audio input signal (10),
B. Determining at least a first, a second and a third time frame of said audio input signal (10) and obtaining a first, a second and a third time frame index parameter (20, 21, 22),
C. Calculating a first, a second and a third signal representation value set (30, 31, 32) for said audio input signal (10) related to each of said first, said second and said third time frame index parameter (20, 21, 22), wherein said first, said second and said third time frame of said audio input signal (10) are matched to a set of frequency channels (40,41,42) respectively, and the calculation of each of said first, said second and said third signal representation value set (30, 31, 32) comprises signal representation parameters for each frequency channel (0-6) of said set of frequency channels (40,41,42), derived from said audio input signal (10) using a windowing function and a short-time Fourier transformation,
D. Calculating a phase derivative value set (50) by phase differentiation for each frequency channel (0-6) of said set of frequency channels (40,41,42) based on said first, said second and said third signal representation value set (30, 31, 32) using finite differences,
E. Integrating said phase derivative value set (50) using a stretching factor (70) and obtaining an output signal representation value set (80),
F. Synthesizing using said output signal representation value set (80) and obtaining an audio output signal (90)

with a corrected phase.

2. A method according to claim 1, wherein said step E of integrating said phase derivative value set (50) is carried out by the steps of:

E1. Adjusting said phase derivative value set (50) and obtaining an adjusted phase derivative value set (60),
E2. Integrating said adjusted phase derivative value set (60) using a stretching factor (70) and obtaining an output signal representation value set (80).

3. A method according to any of the preceding claims, wherein said step E of integrating said phase derivative value set (50) is performed through a phase gradient heap integration algorithm, preferably through a real-time phase gradient heap integration algorithm.

4. A method according to any of the preceding claims, further including, subsequent to said step F of synthesizing, the step of:

G. Resampling of said audio output signal (90), wherein

said first, said second and/or said third time frame index parameter (20, 21, 22) constitute a first time-base of said audio input signal (10), and
said stretching factor (70) constitutes a second time-base of said audio output signal (90), and
said audio output signal (90) is resampled using said first time-base and obtaining a resampled audio signal with pitch-shifted content.

5. A method according to claim 4, wherein at said step G of resampling, a polynomial interpolation is used to obtain said resampled audio signal, and preferably, further an anti-aliasing filter is used to obtain said resampled audio signal.

6. A method according to any of the preceding claims, wherein said phase derivative comprises a time domain partial derivative (35) and a frequency domain partial derivative (36) for each frequency channel (0-6) of said set of frequency channels (40,41,42), respectively.

7. A method according to claim 6, wherein said time domain partial derivative (35) is obtained by using said first, said second and said third signal representation value set (30, 31, 32).

8. A method according to claim 6 or 7, wherein said frequency domain partial derivative (36) is obtained by using frequency channels (0-6) adjacent to each other of said set of frequency channels (40,41,42) from said second signal representation value set (31) with the same time frame index parameter (21).

9. A method according to any of the preceding claims, wherein said number of frequency channels of said set of frequency channels (40,41,42) is set to a value between 512 channels and 16368 channels, preferably between 4096 channels and 16368 channels.

10. A method according to any of the preceding claims, wherein at said step B of determining a signal representation value set (30) from said audio input signal (10), said windowing function covers a time range of 40 ms and 160 ms of said audio input signal, preferably 90 ms.

11. A method according to any of the preceding claims, wherein at said step B of determining a signal representation value set (30) from said audio input signal (10), said windowing function is a Hann window, preferably with a sample size set to a value between 512 samples and 16368 samples, more preferably between 2048 samples and 8192 samples.

12. A method according to any of the preceding claims, wherein said stretching factor (70) is set to a value between 10 and 1/5.

13. A method according to any of the preceding claims, wherein at said step F of synthesizing, a synthesis time step is set to a value between 256 and 16368, preferably between 512 and 2048.

14. A method according to any of the preceding claims, wherein the received audio input signal (10) is non-stationary

and preferably with non-periodic characteristics.

**15.** Device (100) comprising a memory (110) for storing first and a second signal representation value set (30, 31, 32), and a program logic (120) for carrying out the method according to any of the preceding claims, and a processor (130) for carrying out the program logic (120) and/or the method according to any of the preceding claims.

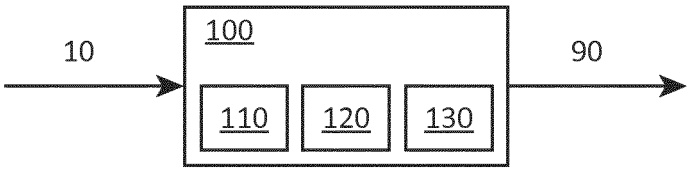

10
100
110
120
130
90

Fig. 1

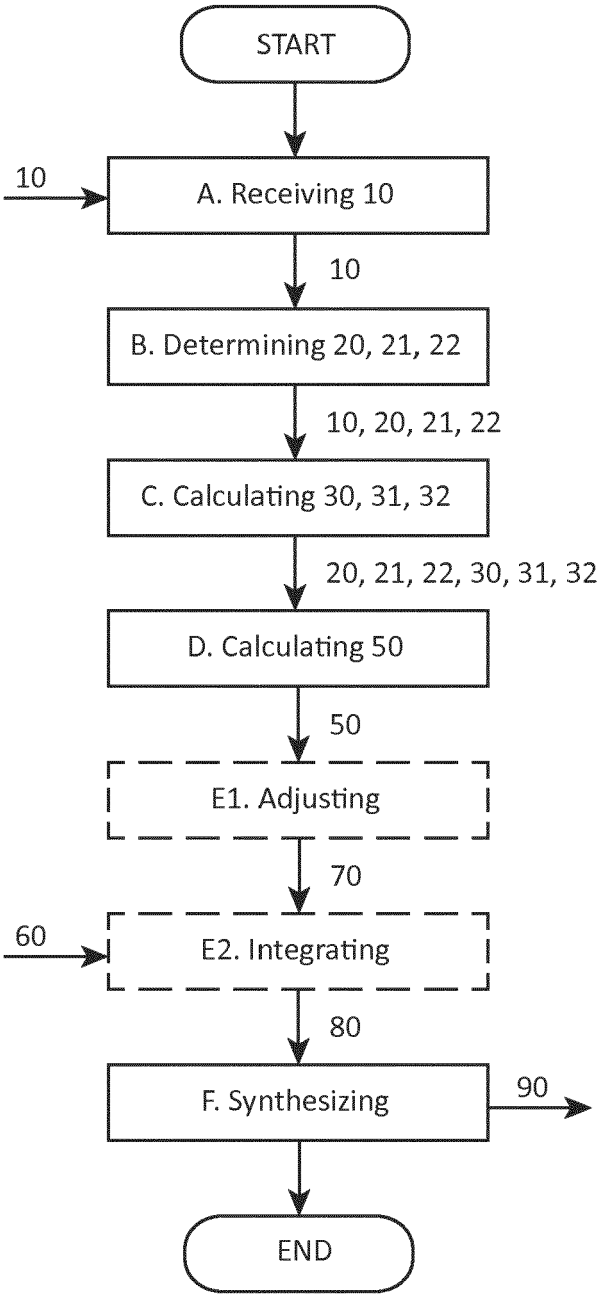

START
10
A. Receiving 10
10
B. Determining 20, 21, 22
10, 20, 21, 22
C. Calculating 30, 31, 32
20, 21, 22, 30, 31, 32
D. Calculating 50
50
E1. Adjusting
70
60
E2. Integrating
80
F. Synthesizing
90
END

Fig. 2

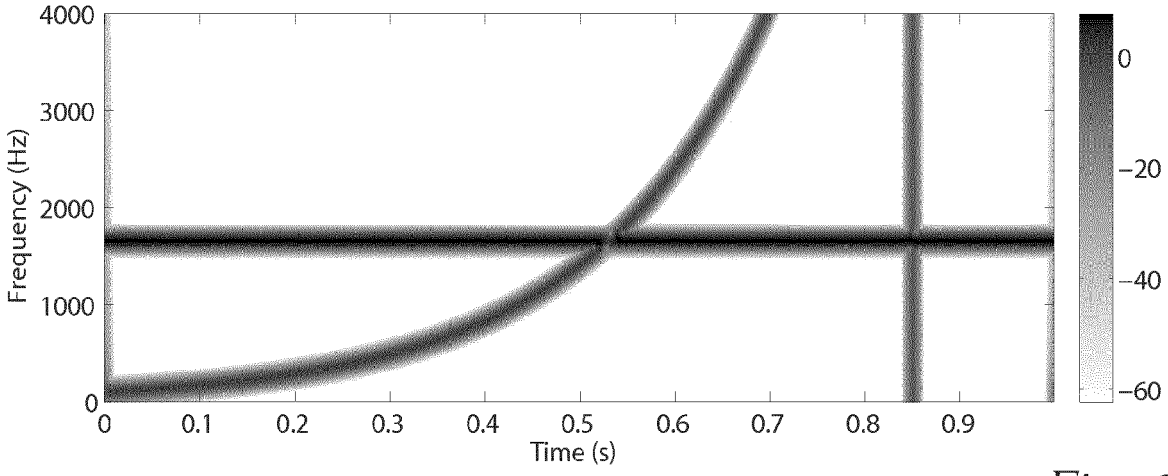



Fig. 3

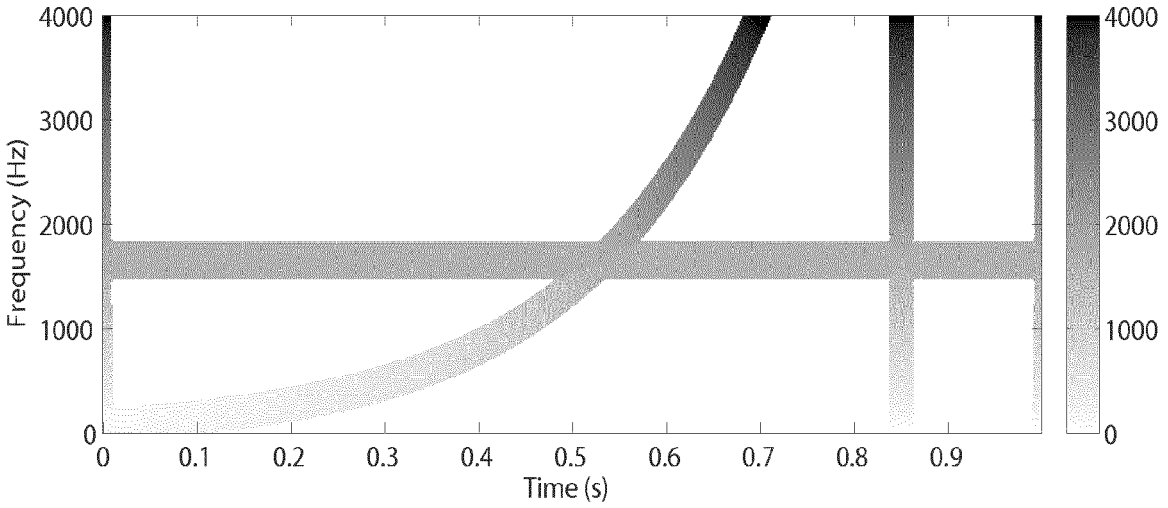



Fig. 4

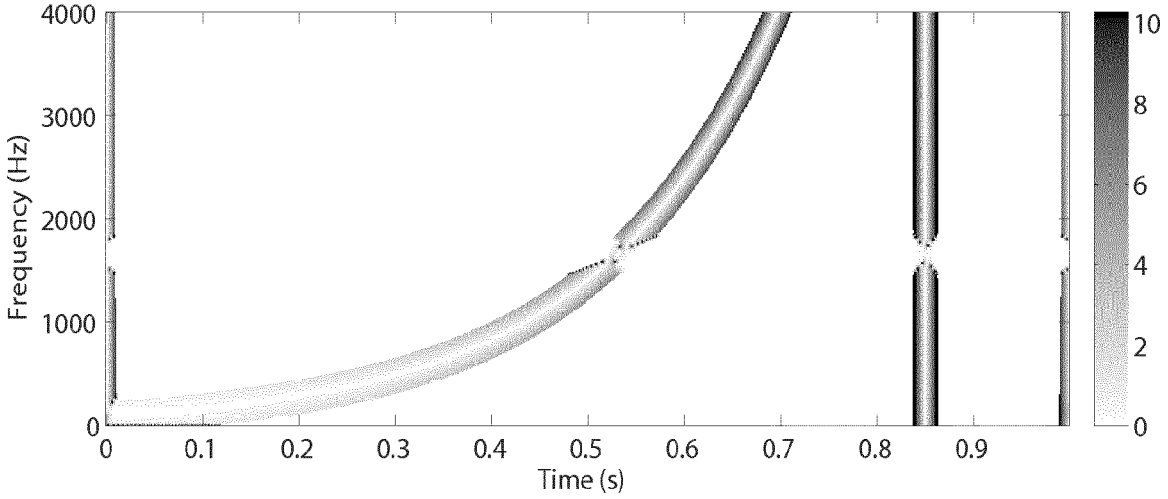



Fig. 5

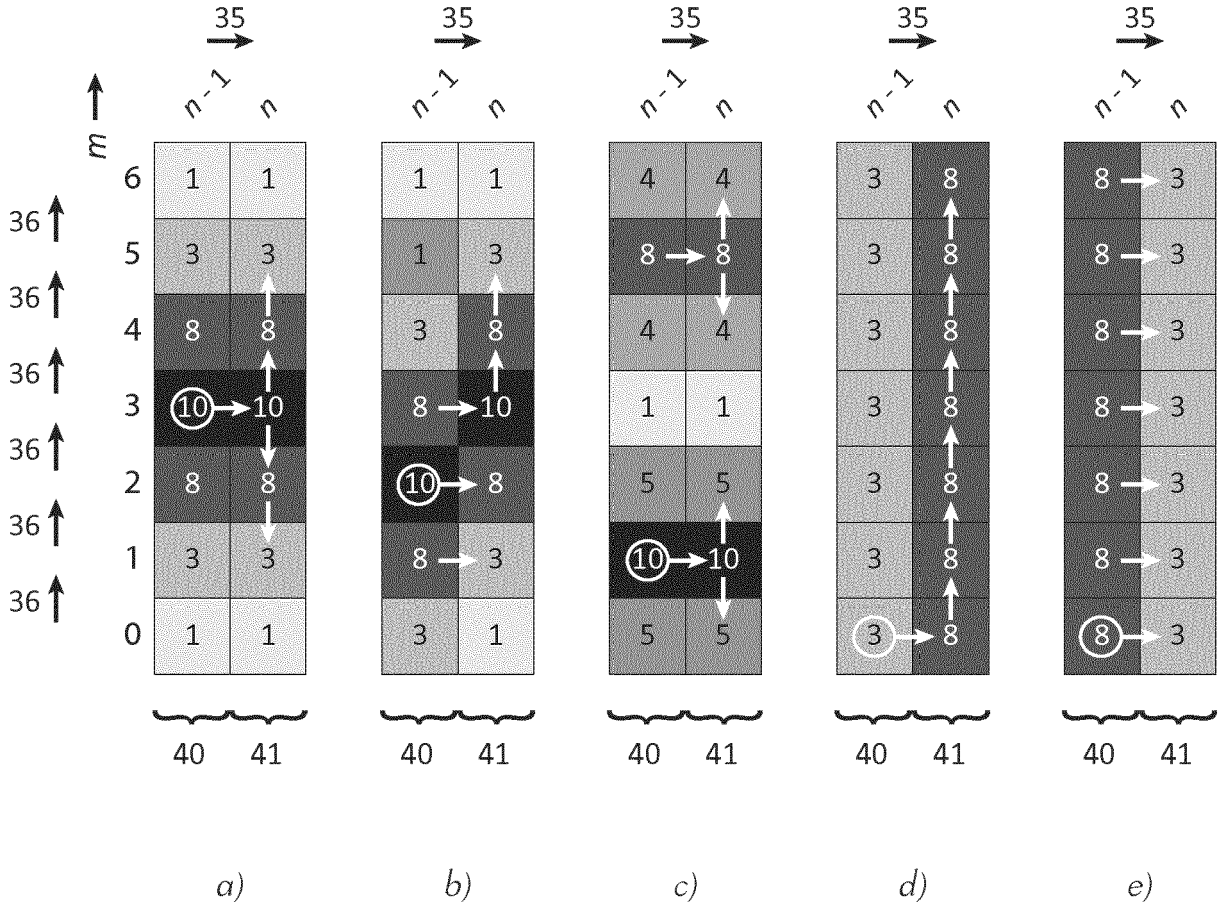
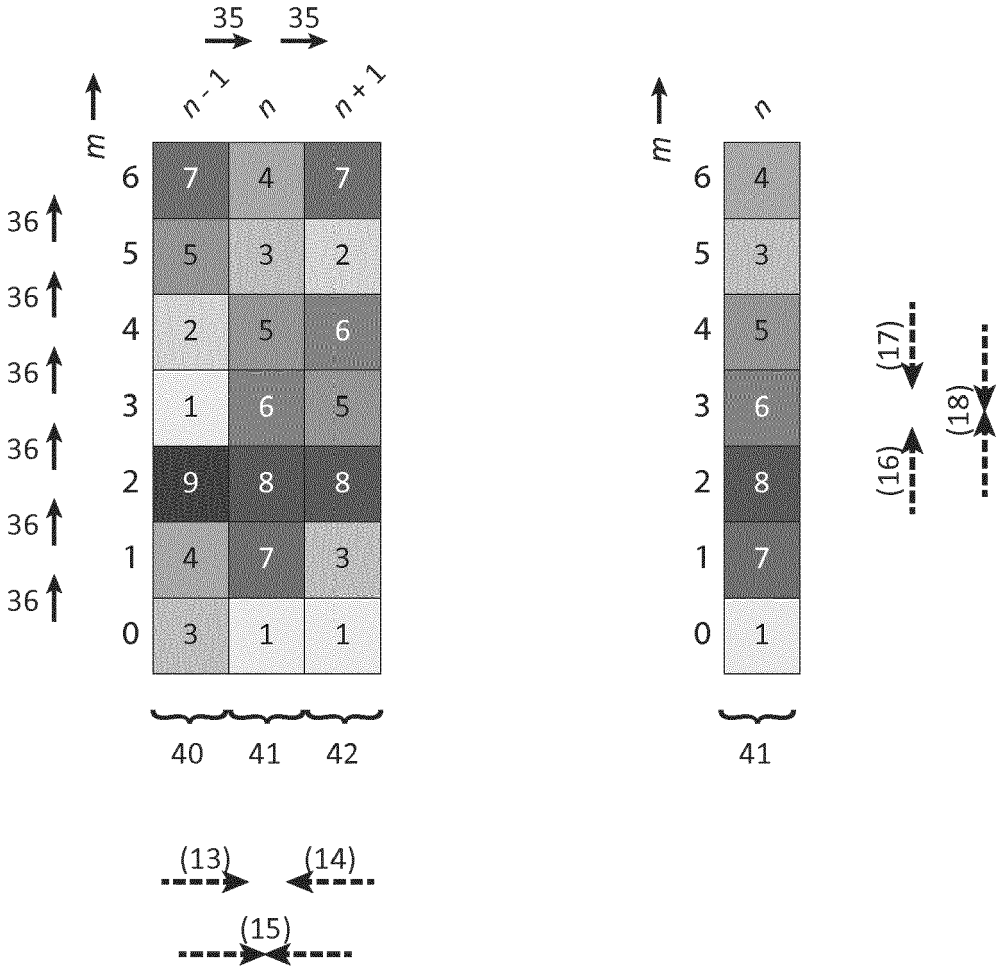

35
n - 1
n
m
6
5
4
3
2
1
0
36
36
36
36
36
36
1 1
3 3
8 8
10 10
8 8
3 3
1 1
40
41
a)
b)
c)
d)
e)

*Fig. 6*


35
35
n - 1
n
n + 1
m
6
5
4
3
2
1
0
7 4 7
5 3 2
2 5 6
1 6 5
9 8 8
4 7 3
3 1 1
36
36
36
36
36
36
40
41
42
(13)
(14)
(15)
m
n
6
5
4
3
2
1
0
4
3
5
6
8
7
1
41
(17)
(18)
(16)

*a)* *b)*

*Fig. 7*

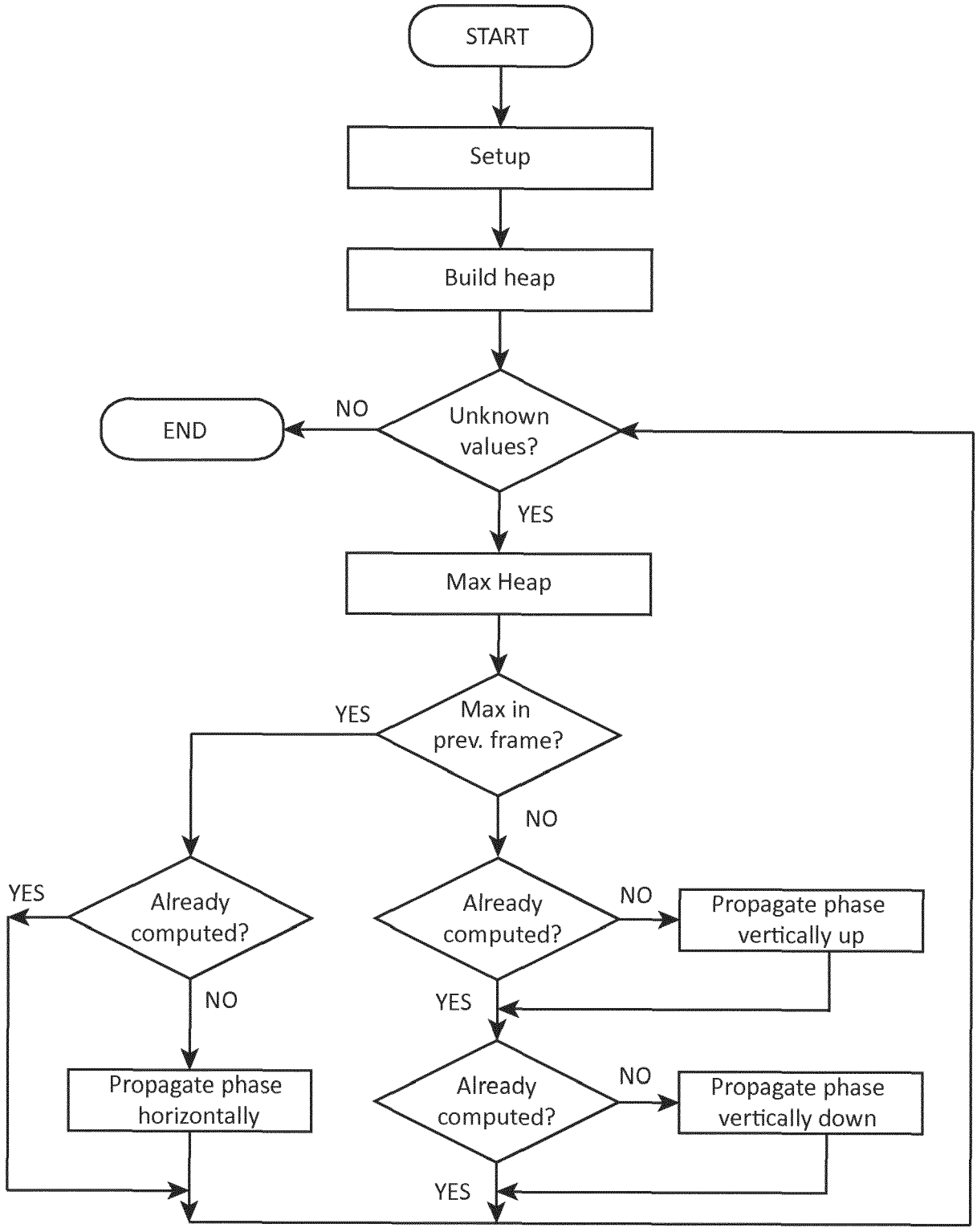

START
Setup
Build heap
Unknown values?
NO
END
YES
Max Heap
Max in prev. frame?
YES
NO
Already computed?
YES
NO
Propagate phase horizontally
Already computed?
NO
Propagate phase vertically up
YES
Already computed?
NO
Propagate phase vertically down
YES

*Fig. 8*

Europäisches Patentamt
European Patent Office
Office européen des brevets

# EUROPEAN SEARCH REPORT

Application Number
EP 17 18 7265

## 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,D | FLANAGAN J L ET AL: "Phase vocoder", BELL SYSTEM TECHNICAL JOURNAL, AT AND T, SHORT HILLS, NY, US, vol. 45, no. 9, 1 November 1966 (1966-11-01), pages 1493-1509, XP011629282, ISSN: 0005-8580, DOI: 10.1002/J.1538-7305.1966.TB01706.X [retrieved on 2014-03-15] | 1,2,9-15 | INV. G10L21/04 G10L19/02 |
| Y | * abstract * * page 1495, line 1 until penultimate paragraph * * section III on pages 1496-1498 * * page 1505, last paragraph * | 4,5 | |
| X,D | JEAN LAROCHE ET AL: "Improved Phase Vocoder Time-Scale Modification of Audio", IEEE TRANSACTIONS ON SPEECH AND AUDIO PROCESSING, IEEE SERVICE CENTER, NEW YORK, NY, US, vol. 7, no. 3, 1 May 1999 (1999-05-01), pages 323-332, XP011054370, ISSN: 1063-6676 * section II.B; page 324, right-hand column - page 325, left-hand column * | 1,2,15 | |
| | -/-- | | |

TECHNICAL FIELDS SEARCHED (IPC)
G10L

The present search report has been drawn up for all claims

| Place of search | Date of completion of the search | Examiner |
|---|---|---|
| Munich | 14 November 2017 | Geißler, Christian |

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)

Europäisches Patentamt
European Patent Office
Office européen des brevets

# EUROPEAN SEARCH REPORT

Application Number
EP 17 18 7265

## 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,D | PRUSA ZDENEK ET AL: "A Noniterative Method for Reconstruction of Phase From STFT Magnitude", IEEE/ACM TRANSACTIONS ON AUDIO, SPEECH, AND LANGUAGE PROCESSING, IEEE, USA, vol. 25, no. 5, 1 May 2017 (2017-05-01), pages 1154-1164, XP011647448, ISSN: 2329-9290, DOI: 10.1109/TASLP.2017.2678166 [retrieved on 2017-04-24] * page 1154, right-hand column, last paragraph * * section IV; page 1157, right-hand column - page 1158, left-hand column * * "Algorithm 1"; page 1158, right-hand column * ----- | 1,3,6-8, 15 | |
| X,D | Zdenek Pr°usa ET AL: "REAL-TIME SPECTROGRAM INVERSION USING PHASE GRADIENT HEAP INTEGRATION", Proceedings of the 19 th International Conference on Digital Audio Effects (DAFx-16), 5 September 2016 (2016-09-05), XP055424074, Retrieved from the Internet: URL:https://ltfat.github.io/notes/ltfatnote043.pdf [retrieved on 2017-11-13] * section 3.2; page 2, right-hand column - page 3, left-hand column * * "Algorithm 1"; page 3, right-hand column * ----- -/-- | 1,3,6,15 | TECHNICAL FIELDS SEARCHED (IPC) |

The present search report has been drawn up for all claims

| Place of search | Date of completion of the search | Examiner |
|---|---|---|
| Munich | 14 November 2017 | Geißler, Christian |

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)

Europäisches Patentamt
European Patent Office
Office européen des brevets

# EUROPEAN SEARCH REPORT

Application Number
EP 17 18 7265

## 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 | TALEB A ET AL: "Partial Spectral Loss Concealment in Transform Coders", 2005 IEEE INTERNATIONAL CONFERENCE ON ACOUSTICS, SPEECH, AND SIGNAL PROCESSING - 18-23 MARCH 2005 - PHILADELPHIA, PA, USA, IEEE, PISCATAWAY, NJ, vol. 3, 18 March 2005 (2005-03-18), pages 185-188, XP010792360, DOI: 10.1109/ICASSP.2005.1415677 ISBN: 978-0-7803-8874-1 * section 3.1; page 186, right-hand column - page 187, left-hand column * * section 4.2; page 187, right-hand column * | 1,15 | |
| Y | US 5 195 166 A (HARDWICK JOHN C [US] ET AL) 16 March 1993 (1993-03-16) * column 7, line 1 - line 13 * | 4,5 | |
| Y | SERRA M H ET AL: "ANALYSIS AND SYNTHESIS OF TONES BY SPECTRAL INTERPOLATION*", JOURNAL OF THE AUDIO ENGINEERING SOCIETY, AUDIO ENGINEERING SOCIETY, NEW YORK, NY, US, vol. 38, no. 3, 1 March 1990 (1990-03-01), pages 111-128, XP000143228, ISSN: 1549-4950 * page 114, right-hand column, last paragraph * * page 124, right-hand column * | 4,5 | TECHNICAL FIELDS SEARCHED (IPC) |

The present search report has been drawn up for all claims

| Place of search | Date of completion of the search | Examiner |
|---|---|---|
| Munich | 14 November 2017 | Geißler, Christian |

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 17 18 7265

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.

14-11-2017

| Patent document cited in search report | Publication date | Patent family member(s) | Publication date |
|---|---|---|---|
| US 5195166 A | 16-03-1993 | AU 658835 B2 | 04-05-1995 |
| | | CA 2091560 A1 | 21-03-1992 |
| | | DE 69131776 D1 | 16-12-1999 |
| | | DE 69131776 T2 | 01-07-2004 |
| | | EP 0549699 A1 | 07-07-1993 |
| | | JP 3467269 B2 | 17-11-2003 |
| | | JP H06503896 A | 28-04-1994 |
| | | KR 100225687 B1 | 15-10-1999 |
| | | US 5195166 A | 16-03-1993 |
| | | US 5226108 A | 06-07-1993 |
| | | US 5581656 A | 03-12-1996 |
| | | WO 9205539 A1 | 02-04-1992 |

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.*

### Non-patent literature cited in the description


- **J. L. FLANAGAN ; R. M. GOLDEN.** Phase vocoder. *Bell System Technical Journal,* 1966, vol. 45 (9), 1493-1509 **[0097]**
- **M. R. PORTNOFF.** Time-scale modification of speech based on short- time Fourier analysis. *Ph.D. dissertation, Massachusetts Institute of Technology,* 1978 **[0097]**
- **E. MOULINES ; J. LAROCHE.** Non-parametric techniques for pitch-scale and time-scale modification of speech. *Speech Communication,* 1995, vol. 16 (2), 175-205 **[0097]**
- **J. LAROCHE ; M. DOLSON.** Phase-vocoder: About this phasiness busi- ness. *IEEE ASSP Workshop on Applications of Signal Processing to Audio and Acoustics,* October 1997, 4 **[0097]**
- Improved phase vocoder time-scale modification of audio. *IEEE Transactions on Speech and Audio Processing,* May 1999, vol. 7 (3), 323-332 **[0097]**
- DAFX: Digital Audio Effects. John Wiley & Sons, Inc, 2002 **[0097]**
- **M. LIUNI ; A. RÖBEL.** Phase vocoder and beyond. *Musical/Tecnologia,* 2013, vol. 7 (0 **[0097]**
- **J. DRIEDGER ; M. MÜLLER.** A review of time-scale modification of music signals. *Applied Sciences,* 2016, vol. 6 (2 **[0097]**
- **C. DUXBURY ; M. DAVIES ; M. B. SANDLER.** Improved time-scaling of musical audio using phase locking at transients. *Audio Engineering Society Convention,* April 2002, vol. 112 **[0097]**
- **A. RÖBEL.** A new approach to transient processing in the phase vocoder. *Proc. Int. Conf. Digital Audio Effects (DAFx-03),* September 2003 **[0097]**
- **E. RAVELLI ; M. SANDLER ; J. P. BELLO.** Fast implementation for non- linear time-scaling of stereo signals. *Proc. Int. Conf. Digital Audio Effects (DAFx-05),* September 2005 **[0097]**
- **F. NAGEL ; A. WALTHER.** A novel transient handling scheme for time stretching algorithms. *Audio Engineering Society Convention,* October 2009, vol. 127 **[0097]**
- **J. DRIEDGER ; M. MÜLLER ; S. EWERT.** Improving time-scale modification of music signals using harmonic-percussive separation. *IEEE Signal Processing Letters,* January 2014, vol. 21 (1), 105-109 **[0097]**
- **E. S. OTTOSEN ; M. DÖRFLER.** A phase vocoder based on nonstationary Gabor frames. *CoRR,* 2016, http://arxiv.org/abs/1612.05156 **[0097]**
- **A. RÖBEL.** A shape-invariant phase vocoder for speech transformation. *Proc. Int. Conf. Digital Audio Effects (DAFx-10),* September 2010, 1-1 **[0097]**
- **A. MOINET ; T. DUTOIT.** PVSOLA: A phase vocoder with synchronized overlap-add. *Proc. Int. Conf. Digital Audio Effects (DAFx-11),* September 2011 **[0097]**
- **T. F. QUATIERI ; R. J. MCAULAY.** Shape invariant time-scale and pitch modification of speech. *IEEE Transactions on Signal Processing,* March 1992, vol. 40 (3), 497-510 **[0097]**
- **D. GRIFFIN ; J. LIM.** Signal estimation from modified short-time Fourier transform. *IEEE Transactions on Acoustics, Speech and Signal Processing,* April 1984, vol. 32 (2), 236-243 **[0097]**
- **Z. PRUŠA ; P. L. SØNDERGAARD.** Real-time spectrogram inversion using phase gradient heap integration. *Proc. Int. Conf. Digital Audio Effects (DAFx-16),* September 2016, 17-21 **[0097]**
- **Z. PRUŠA ; P. RAJMIC.** Toward high-quality real-time signal reconstruction from STFT magnitude. *IEEE Signal Processing Letters,* 06 June 2017, vol. 24 **[0097]**
- **Z. PRUŠA, P. BALAZS ; P. L. SØNDERGAARD.** A Noniterative Method for Reconstruction of Phase from STFT Magnitude. *IEEE/ACM Trans. on Audio, Speech, and Lang. Process.,* May 2017, vol. 25 (5 **[0097]**
- **S. KRAFT ; M. HOLTERS ; A. VON DEM KNESEBECK ; U. ZÖLZER.** Improved PVSOLA time-stretching and pitch-shifting for polyphonic audio. *Proc. Int. Conf. Digital Audio Effects (DAFx-12),* September 2012 **[0097]**
- **J. W. J. WILLIAMS.** Algorithm 232: Heapsort. *Communications of the ACM,* 1964, vol. 7 (6), 347-348 **[0097]**