



(51) International Patent Classification:

G16H 20/30 (2018.01) G10L 17/26 (2013.01)

A61B 5/08 (2006.01) G10L 25/03 (2013.01)

A63B 23/18 (2006.01) G10L 25/66 (2013.01)

(21) International Application Number:

PCT/CA2020/050589

(22) International Filing Date:

01 May 2020 (01.05.2020)

(25) Filing Language:

English

(26) Publication Language:

English

(30) Priority Data:

62/842,019 02 May 2019 (02.05.2019) US

(71) Applicant: MOON FACTORY INC. [CA/CA]; 478-rue St-Ambroise, Montreal, Québec H4C 2C7 (CA).

(72) Inventors: PYLYPENKO, Irene; 1968 rue Gauthier, Montreal, Québec H2K 1A5 (CA). TEKELIOGLU, Osman Zeki; 6673 Av Christophe Colomb, Montreal, Québec H2S 2H1 (CA). DENAULT, Anne-Marie; 478-4020 St-Ambroise, Montreal, Québec H4C 2C7 (CA).

(74) Agent: BENOIT&COTE INC.; 300-560 Cremazie Blvd East, Montreal, Québec H2P 1E8 (CA).

(81) Designated States (unless otherwise indicated, for every kind of national protection available): AE, AG, AL, AM,

AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DJ, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, JO, JP, KE, KG, KH, KN, KP, KR, KW, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, WS, ZA, ZM, ZW.

(84) Designated States (unless otherwise indicated, for every kind of regional protection available): ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, ST, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG).

Published:

— with international search report (Art. 21(3))

(54) Title: SYSTEM FOR MEASURING BREATH AND FOR ADAPTING BREATH EXERCISES

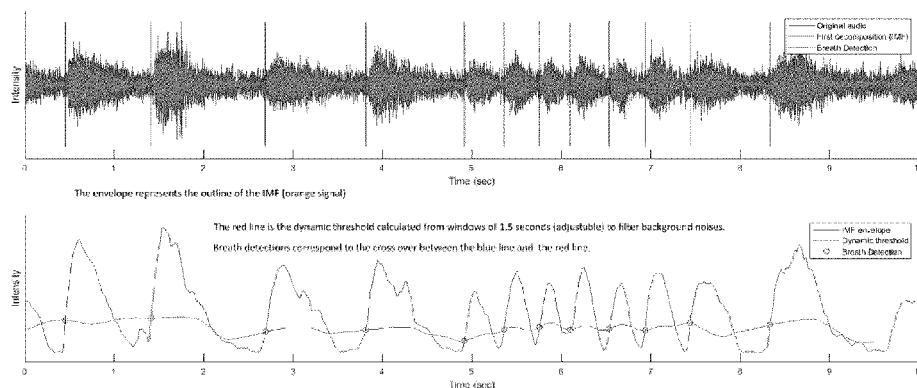


FIGURE 8

(57) **Abstract:** There is described a method for coaching a breathing exercise to a user using a computing device. The method provides a sequence of user instructions for the breathing exercise and determines an expected pattern of breathing intensity from the user in accordance with the sequence of user instructions. Raw audio data stream is retrieved from a microphone associated to the computing device. The raw audio data stream is processed, and breathing events are then detected. By comparing a timing of the detected breathing events with the expected pattern of breathing intensity, compliance with the breathing exercise and future exercises can be adapted within a session or over the length of a program.



SYSTEM FOR MEASURING BREATH AND FOR ADAPTING BREATH EXERCISES

CROSS-REFERENCE TO RELATED APPLICATIONS

[0001] This application claims priority or benefit from U.S. patent application 62/842,019, filed May 2, 2019, the specification of which is hereby incorporated herein by reference in its entirety.

BACKGROUND

(a) Field

[0002] The subject matter disclosed generally relates to systems acting as a personal coach. More specifically, it relates to a system for coaching breath exercises.

(b) Related Prior Art

[0003] Yogic breathing techniques have been developed in India over thousands of years and restricted to exclusive groups, such as yogis and Hindu priests. More recently, as many ancient texts have become widely available, and eastern knowledge has spread throughout the west, more and more people are being trained in these practices. In addition, scientific research has shown the benefits of breathing exercises for many medical conditions and overall wellbeing.

[0004] There is a need to make breathing exercises available to a wider audience while ensuring that the breathing exercises are performed properly and making sure that the breathing exercises undertaken by people are at an appropriate level.

SUMMARY

[0005] In an aspect of the invention, there is provided a method for coaching a breathing exercise to a user using a computing device, the method comprising:

- providing a sequence of user instructions for the breathing exercise;
- determining an expected pattern of breathing intensity from the user in accordance with the sequence of user instructions;
- retrieving a raw audio data stream from a microphone associated to the computing device;
- from the raw audio data stream, detecting breathing events; and
- comparing a timing of the detected breathing events with the expected pattern of breathing intensity to determine compliance with the breathing exercise.

[0006] According to an embodiment, the computing device comprises an operating system which preprocesses the raw audio data stream from the microphone by default for access to the preprocessed

raw audio data stream by any application installed on the computing device, wherein the step of retrieving the raw audio data stream comprises using OS-specific instructions directed to the operating system to avoid receiving the preprocessed raw audio data stream and retrieving the raw audio data stream, not preprocessed, directly, by preventing any alteration by the operating system on the raw audio data stream before retrieval.

[0007] According to an embodiment, the microphone comprises an associated logic module which preprocesses the raw audio data stream from the microphone by default for access to the preprocessed raw audio data stream by any application installed on the computing device, wherein the step of retrieving the raw audio data stream comprises using instructions directed to the associated logic module of the microphone to avoid receiving the preprocessed raw audio data stream and retrieving the raw audio data stream, not preprocessed, directly, by preventing any alteration by the logic module of the microphone on the raw audio data stream before retrieval.

[0008] According to an embodiment, detecting breathing events from the raw audio data stream comprises detecting that a formatted sound intensity has crossed over a threshold, wherein the threshold is based on a moving averaging window of the formatted sound intensity of a duration between 1 and 2s.

According to an embodiment, the raw audio data stream is transformed into a formatted raw audio data stream by dividing the raw audio data stream in independently treated sample windows of a duration between 50ms and 200ms; and converting the raw audio data stream from 16-bit to double.

[0009] According to an embodiment, the formatted raw audio data stream is transformed into the formatted sound intensity by decomposing the formatted raw audio data stream into Intrinsic Mode Functions using Empirical Mode Decomposition.

[0010] According to an embodiment, detecting breathing events comprises:

- determining the threshold for each of the sample windows based on the moving averaging window and identifying intersections of the threshold with an envelope of the Intrinsic Mode Functions;
- for each of the sample windows, calculating an average standard deviation and the threshold to discriminate between a start and an end of a breath event; and
- recording and time-stamping each start of a breath event.

[0011] According to an embodiment, there are further provided the steps of:

- based on the comparing of the timing of the detected breathing events with the expected pattern of breathing intensity to determine compliance with the breathing exercise, generating a score indicative of the compliance; and

- among a library of possible breathing exercises, automatically proposing a future breathing exercise with an adjusted level of difficulty which is based on a level of difficulty of the breathing exercise which was performed and on the score.

[0012] According to an embodiment, there is further provided the step of using a game engine, animating an animated character according to the expected pattern of breathing intensity in timing with the instructions, and displaying the character on the computing device.

[0013] According to an embodiment, detecting breathing events comprises integrating a formatted sound intensity over a continuous period of time covering a detected breath event to evaluate a volume of air that was inspired or expired during the detected breath event.

[0014] According to an embodiment, there is further provided the step of comparing a formatted sound intensity over a continuous period of time with the expected pattern of breathing intensity.

[0015] According to an embodiment, there is further provided the step of adapting future instructions of the breathing exercise based on the step of comparing such that if the step of comparing indicates a low compliance with the expected pattern of breathing intensity, a level of difficulty of the future instructions is lowered within a single session of the breathing exercise.

[0016] According to an embodiment, there are further provided the steps of:

- using a game engine, animating an animated character according to the expected pattern of breathing intensity in timing with the instructions, and displaying the animated character on the computing device; and
- using the same game engine, displaying a plot of the formatted sound intensity in comparison with the expected pattern of breathing intensity over the continuous period of time simultaneously with the animated character or at an end of presentation of the animated character.

[0017] According to another aspect, there is provided a computing device for coaching a breathing exercise to a user, the computing device comprising a microphone associated thereto, a memory storing instructions and a processor executing the instructions to execute steps of:

- providing a sequence of user instructions for the breathing exercise;
- determining an expected pattern of breathing intensity from the user in accordance with the sequence of user instructions;
- retrieving a raw audio data stream from the microphone;
- from the raw audio data stream, detecting breathing events; and

- comparing a timing of the detected breathing events with the expected pattern of breathing intensity to determine compliance with the breathing exercise.

[0018] According to an embodiment, the computing device comprises an operating system which preprocesses the raw audio data stream from the microphone by default for access to the preprocessed raw audio data stream by any application installed on the computing device, wherein the step of retrieving the raw audio data stream comprises using OS-specific instructions directed to the operating system to avoid receiving the preprocessed raw audio data stream and retrieving the raw audio data stream, not preprocessed, directly, by preventing any alteration by the operating system on the raw audio data stream before retrieval.

[0019] According to an embodiment, detecting breathing events from the raw audio data stream comprises detecting that a formatted sound intensity has crossed over a threshold, the threshold based on a moving averaging window of the formatted sound intensity of a duration between 1s and 2s.

[0020] According to an embodiment, the raw audio data stream is transformed into a formatted raw audio data stream by:

- dividing the raw audio data stream in independently treated sample windows of a duration between 50ms and 200ms; and
- converting the raw audio data stream from 16-bit to double.

[0021] According to an embodiment, the formatted raw audio data stream is transformed into the formatted sound intensity by decomposing the formatted raw audio data stream into Intrinsic Mode Functions using Empirical Mode Decomposition.

[0022] According to an embodiment, detecting breathing events comprises:

- determining the threshold for each of the sample windows based on the moving averaging window and identifying intersections of the threshold with an envelope of the Intrinsic Mode Functions;
- for each of the sample windows, calculating an average standard deviation and the threshold to discriminate between a start and an end of a breath event; and
- recording and time-stamping each start of a breath event.

[0023] According to an embodiment, there are further provided the steps of:

- based on the comparing of the timing of the detected breathing events with the expected pattern of breathing intensity to determine compliance with the breathing exercise, generating a score indicative of the compliance; and

- among a library of possible breathing exercises, automatically proposing a future breathing exercise with an adjusted level of difficulty which is based on a level of difficulty of the breathing exercise which was performed and on the score.

[0024] According to an embodiment, there is further provided the step of comparing a formatted sound intensity over a continuous period of time with the expected pattern of breathing intensity.

[0025] According to an embodiment, there are further provided the steps of:

- using a game engine, animating an animated character according to the expected pattern of breathing intensity in timing with the instructions, and displaying the animated character on the computing device; and

- using the same game engine, displaying a plot of the formatted sound intensity in comparison with the expected pattern of breathing intensity over the continuous period of time simultaneously with the animated character or at an end of presentation of the animated character.

[0026] As will be realized, the subject matter disclosed and claimed is capable of modifications in various respects, all without departing from the scope of the claims. Accordingly, the drawings and the description are to be regarded as illustrative in nature, and not as restrictive and the full scope of the subject matter is set forth in the claims.

BRIEF DESCRIPTION OF THE DRAWINGS

[0027] Further features and advantages of the present disclosure will become apparent from the following detailed description of selected embodiments, taken in combination with the appended drawings, in which:

[0028] Figure 1 is a schematic diagram illustrating statuses of a breath coach and their possible transitions, according to an embodiment;

[0029] Figure 2 is a flowchart diagram illustrating a pseudocode generating instructions for coaching breath exercises, including repetitions and transitions between statuses, according to an embodiment;

[0030] Figure 3 is a graph illustrating lung capacity vs. time for different statuses of a breath coach and their possible transitions, according to an embodiment;

[0031] Figure 4 is a schematic diagram illustrating a schematic timeline of breath events to be instructed in a given breath exercise, according to an embodiment;

[0032] Figure 5 is a schematic diagram illustrating data input, transformation and graphical output, according to an embodiment;

[0033] Figure 6 is a schematic diagram illustrating data input, transformation and graphical output, including details of signal processing and graphical production, according to an embodiment;

[0034] Figure 7 is a graph illustrating raw sound data, standard deviation of IMF, corrected standard deviation of IMF and dynamic threshold, according to an embodiment;

[0035] Figure 8 is a graph illustrating (1) raw sound data, IMF, and breath events and (2) envelope of IMF and dynamic threshold with breath events detected on the graph, according to an embodiment;

[0036] Figure 9 is a flowchart diagram illustrating a pseudocode generating instructions for detecting breath events in a data set transformed by EMD into IMF, according to an embodiment;

[0037] Figure 10 is a graph illustrating an envelope of IMF of a user, as measured, compared to an ideal benchmark, according to an embodiment;

[0038] Figure 11 is a pair of graphs illustrating a raw signal and its corresponding envelope of IMF of a user, as measured, which is integrated to evaluate volumetric air flow from and into the lungs, according to an embodiment;

[0039] Figure 12 is a schematic diagram illustrating a user interface for selecting a type of breath exercise, displaying an animated character and displaying a score indicative of the compliance of the user with the instructions, according to an embodiment; and

[0040] Figure 13 is a schematic diagram illustrating an electronic device having a microphone and a display for displaying a graphical user interface which is tactile, with various possible interfaces for recommending and selecting a type of breath exercise, displaying an animated character and displaying a calendar with an exercise program, according to an embodiment.

[0041] It will be noted that throughout the appended drawings, like features are identified by like reference numerals.

DETAILED DESCRIPTION

[0042] The system described herein is intended to make breathing exercises more accessible to a wider audience by using an electronic device to improve adherence of the user to the instructions of a breathing exercise, and to adapt the breathing exercise being coached to the actual breathing pattern of the user, which is being measured in real time.

[0043] To put it briefly, each breathing technique is encoded into an internal computer representation and an associated algorithm, so that each breathing exercise can be translated into instructions that are implemented and adapted into a computer system. According to an embodiment, an animated coach can read these instructions and teach them to the user, wherein the animation of the coach character on a display is adapted to the breathing exercise being instructed, and optionally further adapted to the feedback from the user, such feedback including breathing patterns from the user and compliance of the measured breathing patterns with the breathing exercise being instructed.

[0044] According to an embodiment, the whole method goes as follows. The virtual coach implemented on the electronic device is instructed to voice out loud instructions which explain the breathing exercise and guide the user through them, giving precise instructions on when and how to inhale, exhale and retain the breath, for example, in accordance with the method shown in Fig. 2, which transitions between states such as those shown in Fig. 1, with the goal of having the user breathe according to a pattern such as the one shown in Fig. 3. There are audio instructions and visual cues by the animation on the electronic device. As the user is practicing the breathing exercise, the microphone within the same electronic device (or paired therewith) is used, most probably necessitating the step of providing a direct access of the installed application to the raw audio data collected by the microphone, which can be done by accessing directly the raw audio stream within the operating system of the device. After doing so, the breath-containing audio stream is recorded and detected using a method of detection which can be illustrated with a graph is being plotted with two lines as shown in Figs. 7 or 8, where breath events are detected, to assess real-life compliance of the user with the instructions which were given by the virtual coach, in real time.

[0045] There is now described an exemplary embodiment of a method for encoding of a breathing technique, in this example, “Kapalabhati” (example only, as any other breathing exercise can be encoded).

[0046] The different required “states” of the breathing exercise are first defined. These states and their relationships form a graph of state possibilities, as illustrated in Fig. 1. As shown in Fig. 1, the “states” (each labelled box in the figure) relates to the animated status of the animated character to be displayed, but should also relate to the desired real-life breathing status of the person carrying out the breathing exercise. Accordingly, in terms of computer implementations, each state is used to define an animation, and the relationship threads together different states according to the exercise pattern.

[0047] The animation is performed by varying the body volume of a character, especially the thorax, according to the instructions (inhalation, exhalation, etc.), which are expected to have an effect on the volume of the upper body. An animated character can therefore be generated by computer and various visual states can be translated in a particular body appearance, with the transitions being also

incorporated into the possible animations. At run time, the displayed animated character has an appearance which matches the exact action being instructed in the breathing exercise.

[0048] According to an exemplary embodiment, this pattern of actions, or sequence of consecutive instructions, is encoded in a pseudocode (flowchart of Fig. 2) which represents the actions to be performed by the computer, this flowchart acting as a template used to combine animation states as per variables given. This allows the automatic creation of multiple variations of this breathing exercise, by adjusting the variables, such as number of rounds, or how long the breath retention should be.

[0049] Moreover, still according to an embodiment, an algorithm creates a timeline of a defined variation of a breathing exercise, which is a combination of audio recordings of instructions, the animations as defined by the computer internal representation of the status (Fig. 1), and the background music. An example of this timeline is illustrated in Fig. 4. Finally, as this timeline is played, a breathing pattern graph is produced, driven by the animation states, as illustrated in Fig. 3. In Fig. 3, the colored boxes match the colored boxes in Fig. 1, so for an example, during Avatar_breathing_deeplnhale, a deep inhalation is shown in the graph (dark blue).

[0050] The data pipeline for live biofeedback analysis is illustrated in Fig. 5. First, the breath data is captured via an audio-detecting microphone which the user carries with him or her, or which is in the vicinity of the user with a sound detection range sufficient to detect the breath of this user, such as any one of the following: 1) Built-in microphone in any portable electronic device (e.g., any smartphone having iOS™, Android™, etc. installed thereon), 2) Bluetooth™ headphones with built-in microphone for any device (e.g., adapted for use by pairing with a smartphone, laptop, etc.), or 3) Headphones with a protruding microphone for any device (e.g., for a smartphone, phone system, laptop, gaming console...).

[0051] Thus, the system is device-agnostic, and can be adapted to any platform. After data acquisition using any of the preceding type of microphone, the breath data is processed with signal-processing algorithms, and optionally visualized on a device of choice (the visualization being shown here to illustrate the underlying method of detection of breath).

[0052] An overview of the devices that can be used to implement the method is illustrated in Fig. 6. A platform-specific bridge is built for a specific device-type or brand (such as iOS™, Android™, PS4™, Xbox™, etc.) in order to capture the raw audio data which should contain the sound of breath.

[0053] A particular issue that needs to be addressed is that microphones or devices comprising a microphone typically filter out the sound of breath to improved sound quality, and that raw audio data is not typically available for use. Indeed, most of the time, when recording audio, the devices have built-in pre-processing functionalities to remove the background noise and improve the quality of the audio. This

pre-processing however also removes the faint signal of the breath. The sound data which is required in the present method is therefore made inaccessible by the operating system which filters out useful raw data. This is especially problematic when using a game engine, such as Unity, by which dynamical applications may be implemented in practice, since when using such tools of development, there is a technical challenge regarding recording the breathing because the specific headphones cannot be selected as the device to use for audio capture, and the recording library of functions of such development tools does not allow any configuration. More specifically, the OS automatically does noise processing, which is a problem since this also may remove any breathing signal.

[0054] Thus, in order to capture the breath recordings, there is a need for the method to go past the pre-processing functions, i.e., to circumvent the pre-processing functions by using device-specific functionality to collect the raw data directly instead. According to an embodiment, a solution is to build a bridge where the game engine relies on code closer to the hardware, and to which device-specific and/or OS-specific instructions can be given to specifically start, process and output the raw data, thus having full control of the data processing. In the iOS example, this means writing native iOS code to leverage the native audio framework “Core Audio AV Foundation” to configure the audio capture settings to bypass automatic audio processing. The platform-specific code is written once per platform (OS-specific instructions) to address its audio recording specific configurations.

[0055] The raw data stream from the audio device can be accessed by using OS-specific instructions which allow accessing directly the raw audio file without the normal, default pre-processing performed automatically by the operating system (OS). Indeed, as mentioned above, it is the OS of the device which filters out noise from the built-in microphone of the device. Each OS should comprise instructions which, when performed by an application installed on the electronic device, allow retrieving the original data stream from the microphone on the device. For example, on an iPhone™, there are instructions which can be implemented in an installed application which instruct the operating system (iOS) to retrieve the raw audio stream of the iPhone’s microphone instead of retrieving the typical OS-preprocessed audio stream. Equivalent methods can be applicable to other devices. In the case of external devices, such as Bluetooth-enabled microphones to be paired with the main electronic device, instructions should be found in the SDK (software development kit) to arrive at a similar result of retrieving the raw audio data stream instead of the pre-processed (noise-filtered) one.

[0056] Once this raw audio data stream is obtained, it is assumed that the so-called “noise” comprises sound from the breath, typically filtered out. Furthermore, this raw data is processed with signal processing algorithms according to the invention, which can be implemented, for example, and without limitation, in C#/C++/Matlab™. These algorithms can be applied to raw data received from any device.

[0057] According to an embodiment, and as shown in Fig. 6, at the next step, a game engine, such as Unity™ game engine, receives the data necessary to visualize the users breathing patterns, for example by plotting a graph. This breath pattern graph can be visualized on any device, such as on: 1) A smart phone (e.g., iOS™, Android™, etc.), 2) A game console (e.g., Xbox™, PS4™, etc.), or 3) AR/VR device (e.g., MS Hololens™, etc.).

[0058] The game engine is not required to produce a graph, although it is easier to use the game engine for this purpose if the graph is to be generated within the application itself. However, the game engine may prove additionally useful if the animated character of the displayed virtual coach (which gives oral instructions through the speaker of the electronic device) is adapted in real time to the measured feedback from the user. Whether the animated character depends from feedback from the user or not, the animated character is animated according to the expected pattern of breathing intensity in timing with the instructions, and displayed on the computing device.

[0059] In order to be able to measure the breath pattern in a useful manner, there needs to be a step of data acquisition with respect to the breathing of the user. According to an embodiment, from a microphone of any suitable electronic device in proximity to the user, the following information is captured with exemplary acquisition parameters.

[0060] First, the microphone is operated to record raw sound data, with the step of providing access to the raw sound data stream, as mentioned above, to avoid using the OS-preprocessed noise-filtered data stream. According to an embodiment, a sample window is used, every sample window being 100 ms long, the sound recording being performed at a sampling frequency of 16 000 Hz encoded in 16 bits. The 16-bit data are then converted to doubles (1600 doubles per window), and if file saving is to be used, this data is saved into a WAVE file. The whole file is then converted to an array of doubles.

[0061] As mentioned above, once recorded, the raw sound data stream can be split into sample windows of a duration comprised between 20ms and 300ms, preferably between 50ms and 200ms, more preferably between 80ms and 120ms, for example 100ms. Once the raw sound data stream is split in a plurality of consecutive and distinct sample windows of this duration, each window can be analyzed independently by applying a method of detection of breath thereto.

[0062] In the case of a live version of analysis, the method of detection of breath is applied directly to each consecutive 100ms-window at run time, in real time as the sound is recorded, without first saving it into a file (the saving into the file of the raw sound stream being therefore only necessary if the breath patterns are to be detected after the recording session).

[0063] In order to apply a method of detection of breath to the raw sound stream, the processor of the computer will execute instructions which cause it to detect features in the raw sound stream using detection parameters such as the following ones.

[0064] First, a moving averaging time window is applied on the consecutive sample windows to determine an average sound level. The deviation of the sound intensity compared to the average sound level, above or below a threshold, is used to determine if the sound includes breathing sound or not. This “threshold averaging time window” is set at a time duration between 0.8s and 3s, preferably between 1s and 2s, and more preferably at about 1.5s.

[0065] A threshold multiplier, between 0.2 and 0.8, preferably between 0.4 and 0.6, preferably set at about 0.5, is used to determine how many times the standard deviation of the signal is representative of the threshold for detecting a breath event in the algorithm for breath detection, e.g., the threshold is $\text{stddev} \times 0.5$. The Breath Detection Threshold Factor represents how high stddev should be relative to moving threshold). There can be set a Breath Detection Minimum Time and a Breath Detection Maximum Time S.

[0066] The data captured is visually represented in Figure 7, with the following color encoding: Purple: Represents the RAW audio signal; Red: Threshold, based on standard deviation with an offset and filtering with moving average on 1.5s; Teal: Standard deviation calculated from IMF (explained below); Blue: Standard deviation shifted on X axis to match threshold and compensate lag.

[0067] In summary, first an EMD (Empirical Mode Decomposition) function is applied to the formatted raw sound data decompose the signal into a number of IMFs (Intrinsic Mode Functions). IMF is an oscillatory function with time-varying frequencies that represent the local characteristics of nonstationary signals. Then, for each sample, the average standard deviation is calculated (with reference to the algorithm below) along with the threshold value. Based on the difference between these two values, each intersection between a signal is detected by the processor. If the intersection is detected and the standard deviation is positive, a breath is detected, otherwise, it is not considered to be a breath event.

[0068] The audio signal is analyzed using the signal amplitude, which is typically measured in decibels. The dynamic threshold (red line below) is used to ignore the background noise from the process. This threshold adjusts itself continuously according to a moving averaging window of a duration of 1.5 seconds (this moving averaging window is configurable within the parameters mentioned above) so if the background noise increases, the red line will also increase on the graph shown in Fig. 7.

[0069] Fig. 8 also shows the raw sound data stream, the stream transformed into IMF, and then the IMF envelope compared to the moving threshold, with breathing events detected in both graphs.

[0070] Fig. 9 illustrates, in the form of a pseudocode flowchart, the steps that are performed for transforming the data, detecting an intersection, and ensuring that the behavior of the standard deviation is indicative of the start of the breath event (or the end thereof), where the start of each breath event is time-stamped and recorded.

[0071] The parameters for the method for detecting breath in a raw data stream (previously formatted as mentioned above) can be fine-tuned with the parameters listed in Table 1.

Table 1 – Parameters used for breath detection (exemplary values)

Parameter	Description
StandardDevRollingAvgFilterSmoothS	Moving average applied over the blue line (and teal line), typically 0.2s. It helps removing undesirable noise spikes not related to breath.
DetectionThresholdSmoothS	1.5s red line average filter is applied. Increasing this time will delay algorithm reactivity and increase CPU usage.
DetectionThresholdMultiplier	Example value: 0.5. Variance is multiplied by this parameter as trigger input data (red line)
BreathDetectionMinHeightThresholdFactor	How much should we go higher than the trigger line?
BreathDetectionMinWidthThresholdS	Minimum time length to be considered breath
BreathDetectionMaxWidthThresholdS	Maximum time of a breath

[0072] The algorithm expects to get as the input a MONO / DOUBLE 8 bytes-based audio signal. Typically, it is bounded -1 to +1. According to an embodiment, and depending on the implementation, it can be important to ensure it is a mono stream, as the algorithm may not be able to decode a stereo stream.

[0073] If this input type is not detected, a conversion is done before entering the algorithm. For example, let us suppose a 16,000Hz sampling rate is used, and a data stream is delivered/inputted to the detection algorithm every 100ms (duration of the sample window), so 1,600 double values are inputted per second. The algorithm can work well at other sampling frequencies, such as 8,000Hz.

[0074] As the output of the detection algorithm, when the algorithm detects new breathing event, it will add it to a list (with a time stamp). Such events are then retrieved or accessed using these functions:

- public IEnumerable<Breath> popBreathingDetections()
- public IEnumerable<Breath> peekBreathingDetections()

[0075] Breath object contains a flag indicating if detection is valid or why it was rejected if it's not valid. It also contains a timestamp. The "pop" variant will remove breathing from the list after being called, and "peek" will return breathing but without removing them from the list.

[0076] According to an embodiment, the breathing intensity is not only detected, but also its intensity is monitored and compared with the expected ideal breathing intensity pattern, as illustrated in Fig. 10, where this comparison with a benchmark can be shown to the user either in real time or a posteriori on the electronic device.

[0077] In order to collect good quality data and reduce background noise, it is important that the user is guided to be in a suitable environment for the breathing session. Before the breathing session begins, the user will be advised to be in a quiet room by themselves, and either wear headphones or have the phone within a maximum distance (range) from the mouth. According to an embodiment, this is followed by a calibration step, where the user is asked to take 3 normal breaths, allowing the system to calibrate the breathing exercise data against this 'normal' data, as shown in Fig. 10. The blue line is the coach's breathing pattern (the model pattern for the user to strive towards), and the red line is the user's actual (measured) breathing pattern, including intensity over a continuous period of time covering the exercise. In the case of Fig. 10, the same game engine as the one used to animate the character is used for displaying a plot of the formatted sound intensity (based on the IMF envelope, for example) in comparison with the expected pattern of breathing intensity over the continuous period of time (either simultaneously with the animated character or at an end of presentation of the animated character).

[0078] The blue line is a function of the coach's inhalations and exhalations, reacting to the body and lung movements of the animation, i.e., consistent with the breathing status, the expected associated breath events and timing as shown in the examples of Figs. 1-3. The red line is the user's actual breathing, plotted from the output of the signal processing algorithm described above.

[0079] According to an embodiment, the lung movement, i.e., the lung volume or capacity over time and/or, equivalently, the volumetric flow of air which is expired/inspired over time, can be evaluated using the present method. In the case of nasal breathing, the base assumption of this evaluation is that the pulmonary flow (air flow) is proportional to the sound amplitude caused by the flow of air through the nostrils. It was found that this is not the case with the flow of air through the mouth, the sound of which can be modulated according to the shape given to the mouth. Also, interestingly, it was found that the audio spectrum of the sound produced by the nostrils during the passage of air can be characterized according to the flow rate, either by a displacement of the dominant frequency or the line appearance for certain air flow rates.

[0080] Following the base assumption, if the amplitude of the sound signal produced by the breath during nasal breathing is proportional to its flow, the integration of the envelope of the signal (area under the curve) should be proportional to the volume of air inhaled or exhaled during a given time. The integration limits correspond to the beginning and the end of each inspiration and expiration. Figure 11 is

a pair of graphs illustrating a raw signal and its corresponding envelope of IMF of a user, as measured, which is integrated to evaluate volumetric air flow from and into the lungs over a period of time corresponding to the duration of a breath event (expiration or inspiration), resulting in the volume expired or inspired during an expiration or an inspiration, respectively, as mentioned above.

[0081] According to an embodiment, a calibration is carried out in order to determine for each user the relationship between the sound level and the air flow. The relationship between noise level and air flow may depend on several other factors that are determined and taken into account for each user. The calibration is based on the maximum lung capacity, i.e. one inspiration and a maximum expiration carried out as quickly as possible, and another over a longer period (4 to 5 sec.). Both measures should, in principle, give the same inspired and expired volume and help correlate the two.

[0082] According to an embodiment, there is further generated a session score to assess the quality of the data. The session score can be used to quantify the quality of the breathing session, representing how closely the user followed the coach. For example, scores can be calculated as follows:

$$\text{sessionScore} = (\text{rhythmScore} + \text{amplitudeScore}) * \text{thresholdFactor}$$

$$\text{rhythmScore} = (\# \text{ peaks} + \# \text{ troughs of user}) / (\# \text{ peaks} + \# \text{ troughs of model}) - \text{avg}(\text{distance between the overlaps})$$

$$\text{amplitudeScore} = \text{sum}(\text{diff}(\text{peaks}) + \text{diff}(\text{troughs}))$$

[0083] Through the mobile user interface, a personalized user experience is then offered to each individual user, based on their own respective preferences. First, a questionnaire may be implemented which asks why they want to practice breathing exercises and/or what objective they want to achieve. Then, the user is asked if they have certain medical conditions that would restrict them to perform the more challenging exercises. Finally, their current experience with breathing exercises is queried.

[0084] Based on the answers to the questionnaire, a personalized program is automatically generated for them, according to the restrictions, goals and previous experience, knowing that particular exercise programs are associated with a certain level of difficulty and a level of required physical engagement.

[0085] Further, as the user practices the exercises over a certain period of time (i.e., a plurality of exercises having been completed and assessed), the program can be automatically based on their past performance. If they are doing exceptionally well, a more challenging program is recommended. If they are underperforming, a more suitable program (i.e., lower level of difficulty) is recommended for the next exercise. The underlying logic is built to adapt to the user's current state, and to offer them exercises that help them grow and reach their objectives.

[0086] A non-limiting example of a pathway that a user may follow is illustrated in Fig. 12. Throughout the onboarding process, as data about the user is collected, they select “Winding down” as their objective. Based on this information, the Destress program is recommended for them. They follow each session practicing the associated breathing exercises. After each session, the computer measures and showcases Session Score, determining how closely the user followed the coach’s instructions. Finally, after 3 sessions, based on the Session Score, we adjust the recommendation, either to continue, try a program to practice a specific skill, such as focusing, or a more difficult program to challenge them further.

[0087] According to an embodiment, the output on the display of the user interface adapts to the live, real-time biofeedback. First, based on the user’s initial profile settings, there is automatically recommended a breathing program to attain the desired objective, based on the user’s experience level and medical restrictions. A breathing program is a sum of individual breathing sessions, where each session is a combination of 3 to 5 breathing exercises that the coach guides the user through.

[0088] Second, as the user starts to follow a session, the coach’s instructions adapt to the user’s performance, changing the length of the session according to the speed of the user’s progress as measured using the breath detection algorithm and following score, evaluated dynamically in real-time during a session. The coach (i.e., the instructions being given to the user during the session) can automatically adapt the length of breath retentions, lengths of inhalations and exhalations as being measured during the session, and the number of repetitions of breathing exercises. For example, if a breathing exercise requires to hold the breath for 15 seconds 5 times, but they can only hold it for 10 seconds the first time, then the coach shall detect the breath events and evaluate the low score of compliance, where the low score of compliance as measured within the session indicates that the exercises are not well adapted to the level of the user, and therefore adapt the remaining 4 retentions to be 10 seconds long, for example, to be more adapted to the real-life level of the user as being measured in real-time.

[0089] Each day the user logs in, the coach can see where the user is that day, and adapt the length of retentions based their personal performance that day, to bring them to the final objective to complete the program.

[0090] Third, the overall objective of the breathing program is to improve the user’s breathing quality and lung capacity, by incrementally giving more difficult exercises. For example, the next session has breath retentions for 5 seconds more than the previous one. The level of difficulty of the exercises is therefore adapted to be more in line with an actual level of the user within a session, and is gradually

increased over successive sessions (i.e., during the program) to force the user to improve in the long term.

[0091] Figure 13 is a schematic diagram illustrating an electronic device having a microphone and a display for displaying a graphical user interface which is tactile, with various possible interfaces for recommending and selecting a type of breath exercise among an available library of exercises, displaying an animated character with a particular esthetics and displaying a calendar with an exercise program.

[0092] The method described above is advantageously implemented on a computing device, typically portable or handheld, such as a smartphone (as shown in Fig. 13, having a built-in microphone and an interactive display screen), a tablet, a phablet, or a web-enable device wearable device (watch, glasses, etc.). Most or all steps are executed locally, by the processor of the device which executes the instructions stored thereon (application installed thereon). Optionally, a few steps such as a posteriori analysis of the data can be performed on a remote server, in which case a web connection is required.

[0093] In all cases, the computing device is essential for carrying out the method, since the sound data needs to be collected in a quantifiable manner only available with computing devices, especially since transformed data sets are to be computed in real time. It follows that the microphone associated to the computing device (either associated permanently or temporarily) is also essential for carrying out the method.

[0094] While preferred embodiments have been described above and illustrated in the accompanying drawings, it will be evident to those skilled in the art that modifications may be made without departing from this disclosure. Such modifications are considered as possible variants comprised in the scope of the disclosure.

CLAIMS:

1. A method for coaching a breathing exercise to a user using a computing device, the method comprising:
 - providing a sequence of user instructions for the breathing exercise;
 - determining an expected pattern of breathing intensity from the user in accordance with the sequence of user instructions;
 - retrieving a raw audio data stream from a microphone associated to the computing device;
 - from the raw audio data stream, detecting breathing events; and
 - comparing a timing of the detected breathing events with the expected pattern of breathing intensity to determine compliance with the breathing exercise.
2. The method of claim 1, wherein the computing device comprises an operating system which preprocesses the raw audio data stream from the microphone by default for access to the preprocessed raw audio data stream by any application installed on the computing device, wherein the step of retrieving the raw audio data stream comprises:
 - using OS-specific instructions directed to the operating system to avoid receiving the preprocessed raw audio data stream and retrieving the raw audio data stream, not preprocessed, directly, by preventing any alteration by the operating system on the raw audio data stream before retrieval.
3. The method of claim 1, wherein the microphone comprises an associated logic module which preprocesses the raw audio data stream from the microphone by default for access to the preprocessed raw audio data stream by any application installed on the computing device, wherein the step of retrieving the raw audio data stream comprises:
 - using instructions directed to the associated logic module of the microphone to avoid receiving the preprocessed raw audio data stream and retrieving the raw audio data stream, not preprocessed, directly, by preventing any alteration by the logic module of the microphone on the raw audio data stream before retrieval.
4. The method of any one of claims 1 to 3, wherein detecting breathing events from the raw audio data stream comprises:

- detecting that a formatted sound intensity has crossed over a threshold, wherein the threshold is based on a moving averaging window of the formatted sound intensity of a duration between 1s and 2s.
5. The method of claim 4, wherein the raw audio data stream is transformed into a formatted raw audio data stream by:
- dividing the raw audio data stream in independently treated sample windows of a duration between 50ms and 200ms; and
 - converting the raw audio data stream from 16-bit to double.
6. The method of claim 5, wherein the formatted raw audio data stream is transformed into the formatted sound intensity by:
- decomposing the formatted raw audio data stream into Intrinsic Mode Functions using Empirical Mode Decomposition.
7. The method of claim 6, wherein detecting breathing events comprises:
- determining the threshold for each of the sample windows based on the moving averaging window and identifying intersections of the threshold with an envelope of the Intrinsic Mode Functions;
 - for each of the sample windows, calculating an average standard deviation and the threshold to discriminate between a start and an end of a breath event; and
 - recording and time-stamping each start of a breath event.
8. The method of any one of claims 1 to 7, further comprising:
- based on the comparing of the timing of the detected breathing events with the expected pattern of breathing intensity to determine compliance with the breathing exercise, generating a score indicative of the compliance; and
 - among a library of possible breathing exercises, automatically proposing a future breathing exercise with an adjusted level of difficulty which is based on a level of difficulty of the breathing exercise which was performed and on the score.
9. The method of any one of claims 1 to 8, further comprising:

- using a game engine, animating an animated character according to the expected pattern of breathing intensity in timing with the instructions, and displaying the character on the computing device.
10. The method of claim 6, wherein detecting breathing events comprises:
- integrating a formatted sound intensity over a continuous period of time covering a detected breath event to evaluate a volume of air that was inspired or expired during the detected breath event.
11. The method of claim 6 or 10, further comprising:
- comparing a formatted sound intensity over a continuous period of time with the expected pattern of breathing intensity.
12. The method of claim 11, further comprising:
- adapting future instructions of the breathing exercise based on the step of comparing such that if the step of comparing indicates a low compliance with the expected pattern of breathing intensity, a level of difficulty of the future instructions is lowered within a single session of the breathing exercise.
13. The method of claim 11, further comprising:
- using a game engine, animating an animated character according to the expected pattern of breathing intensity in timing with the instructions, and displaying the animated character on the computing device; and
 - using the same game engine, displaying a plot of the formatted sound intensity in comparison with the expected pattern of breathing intensity over the continuous period of time simultaneously with the animated character or at an end of presentation of the animated character.
14. A computing device for coaching a breathing exercise to a user, the computing device comprising a microphone associated thereto, a memory storing instructions and a processor executing the instructions to execute steps of:
- providing a sequence of user instructions for the breathing exercise;
 - determining an expected pattern of breathing intensity from the user in accordance with the sequence of user instructions;

- retrieving a raw audio data stream from the microphone;
- from the raw audio data stream, detecting breathing events; and
- comparing a timing of the detected breathing events with the expected pattern of breathing intensity to determine compliance with the breathing exercise.

15. The computing device of claim 14, wherein the computing device comprises an operating system which preprocesses the raw audio data stream from the microphone by default for access to the preprocessed raw audio data stream by any application installed on the computing device, wherein the step of retrieving the raw audio data stream comprises:

- using OS-specific instructions directed to the operating system to avoid receiving the preprocessed raw audio data stream and retrieving the raw audio data stream, not preprocessed, directly, by preventing any alteration by the operating system on the raw audio data stream before retrieval.

16. The computing device of any one of claims 14 to 15, wherein detecting breathing events from the raw audio data stream comprises:

- detecting that a formatted sound intensity has crossed over a threshold, wherein the threshold is based on a moving averaging window of the formatted sound intensity of a duration between 1s and 2s.

17. The computing device of claim 16, wherein the raw audio data stream is transformed into a formatted raw audio data stream by:

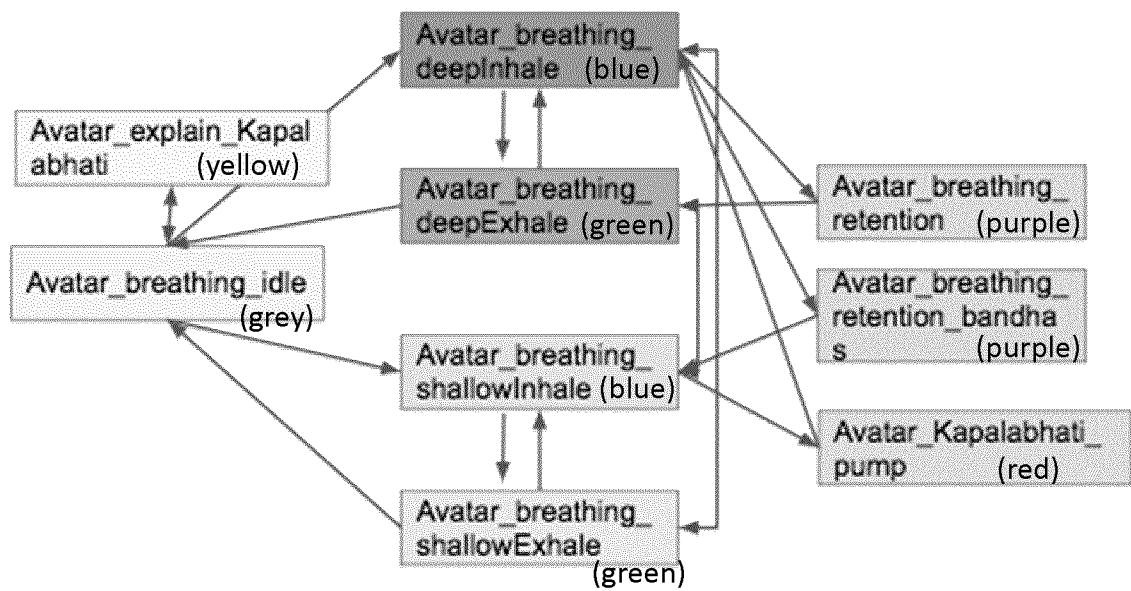
- dividing the raw audio data stream in independently treated sample windows of a duration between 50ms and 200ms; and
- converting the raw audio data stream from 16-bit to double.

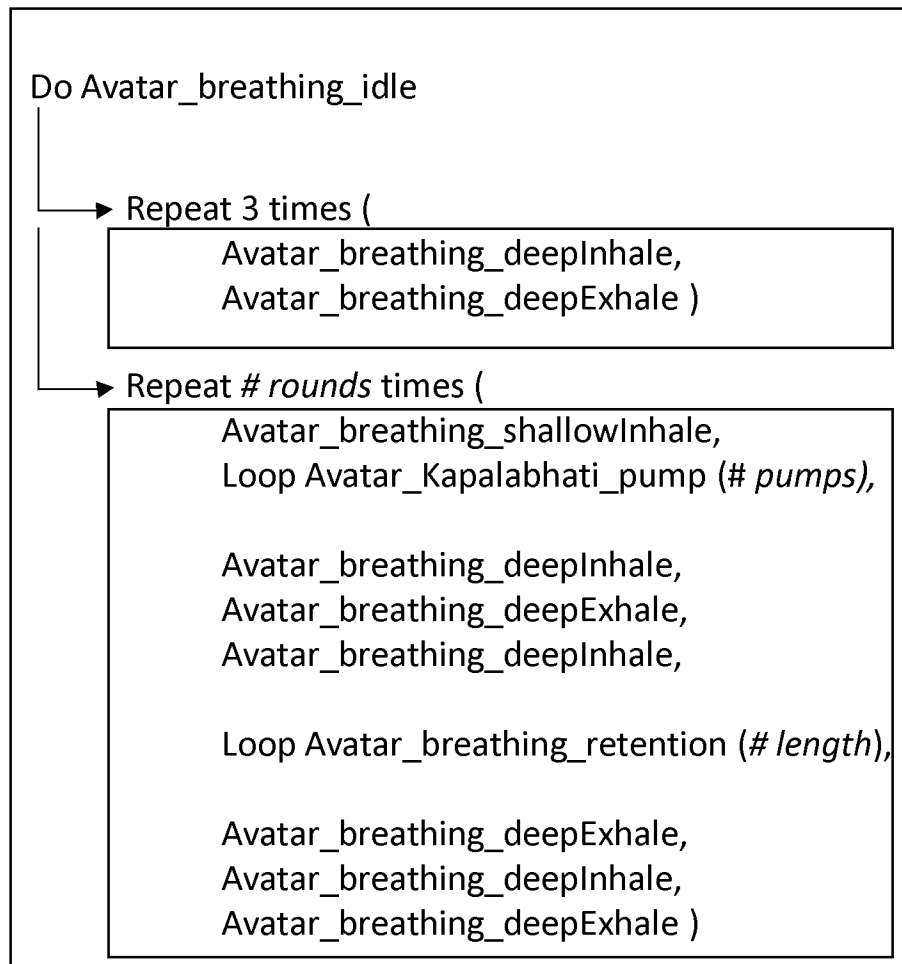
18. The computing device of claim 17, wherein the formatted raw audio data stream is transformed into the formatted sound intensity by:

- decomposing the formatted raw audio data stream into Intrinsic Mode Functions using Empirical Mode Decomposition.

19. The computing device of claim 18, wherein detecting breathing events comprises:

- determining the threshold for each of the sample windows based on the moving averaging window and identifying intersections of the threshold with an envelope of the Intrinsic Mode Functions;
 - for each of the sample windows, calculating an average standard deviation and the threshold to discriminate between a start and an end of a breath event; and
 - recording and time-stamping each start of a breath event.
20. The computing device of any one of claims 14 to 19, further comprising:
- based on the comparing of the timing of the detected breathing events with the expected pattern of breathing intensity to determine compliance with the breathing exercise, generating a score indicative of the compliance; and
 - among a library of possible breathing exercises, automatically proposing a future breathing exercise with an adjusted level of difficulty which is based on a level of difficulty of the breathing exercise which was performed and on the score.
21. The computing device of claim 18, further comprising:
- comparing a formatted sound intensity over a continuous period of time with the expected pattern of breathing intensity.
22. The computing device of claim 21, further comprising:
- using a game engine, animating an animated character according to the expected pattern of breathing intensity in timing with the instructions, and displaying the animated character on the computing device; and
 - using the same game engine, displaying a plot of the formatted sound intensity in comparison with the expected pattern of breathing intensity over the continuous period of time simultaneously with the animated character or at an end of presentation of the animated character.

**FIGURE 1**

**Input Variables**

- retention: 15-120 seconds (min-max)
- pumps: 15-200 (min-max)
- rounds: 1-3(min-max)

FIGURE 2

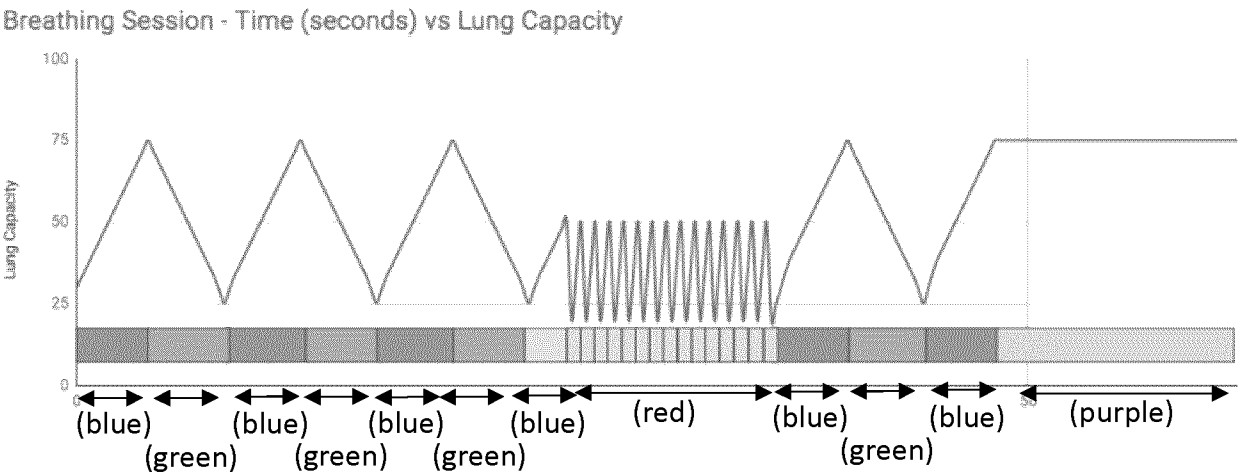


FIGURE 3

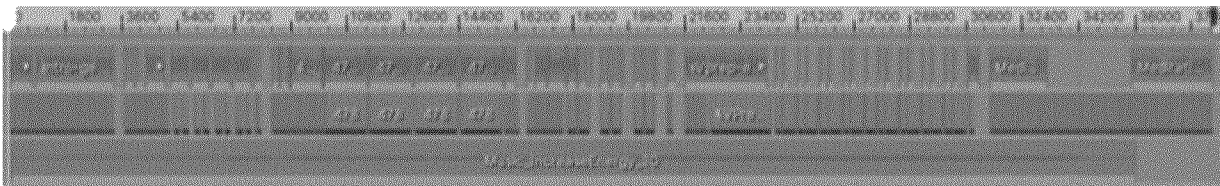
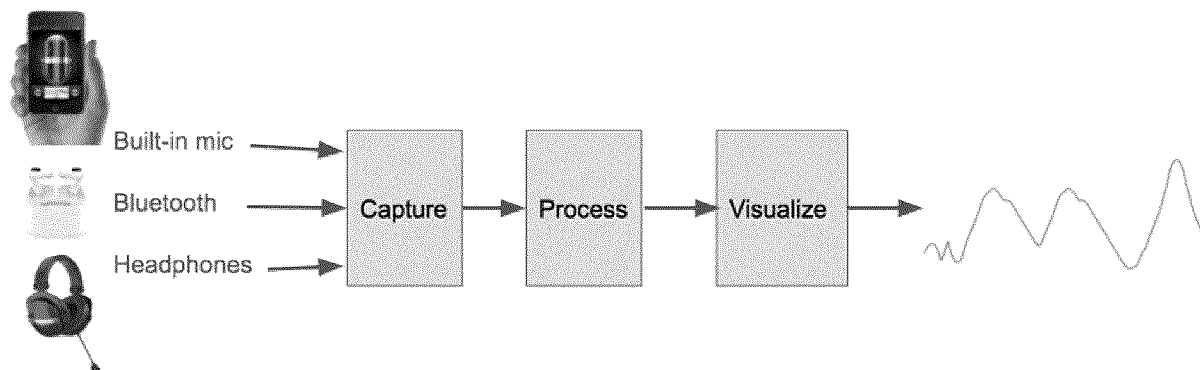
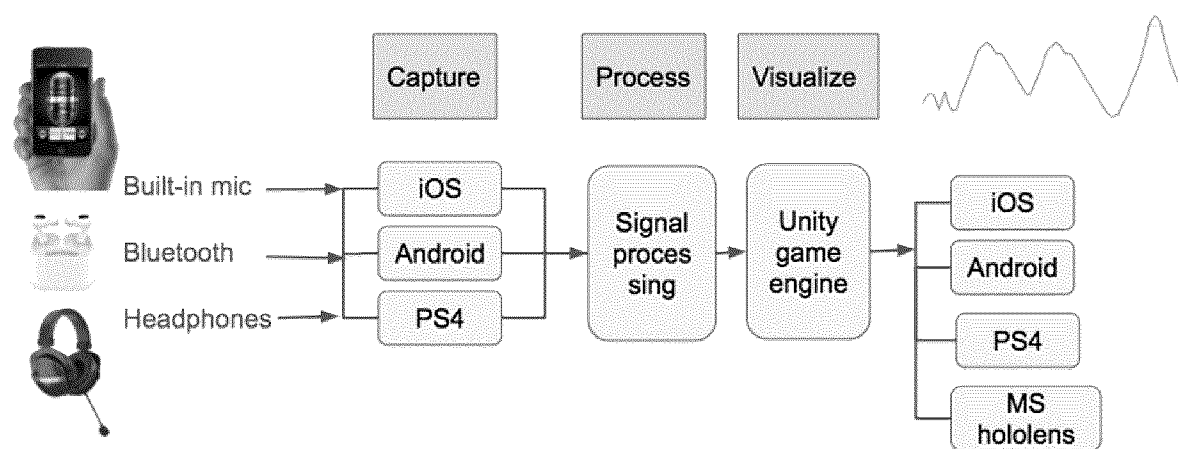
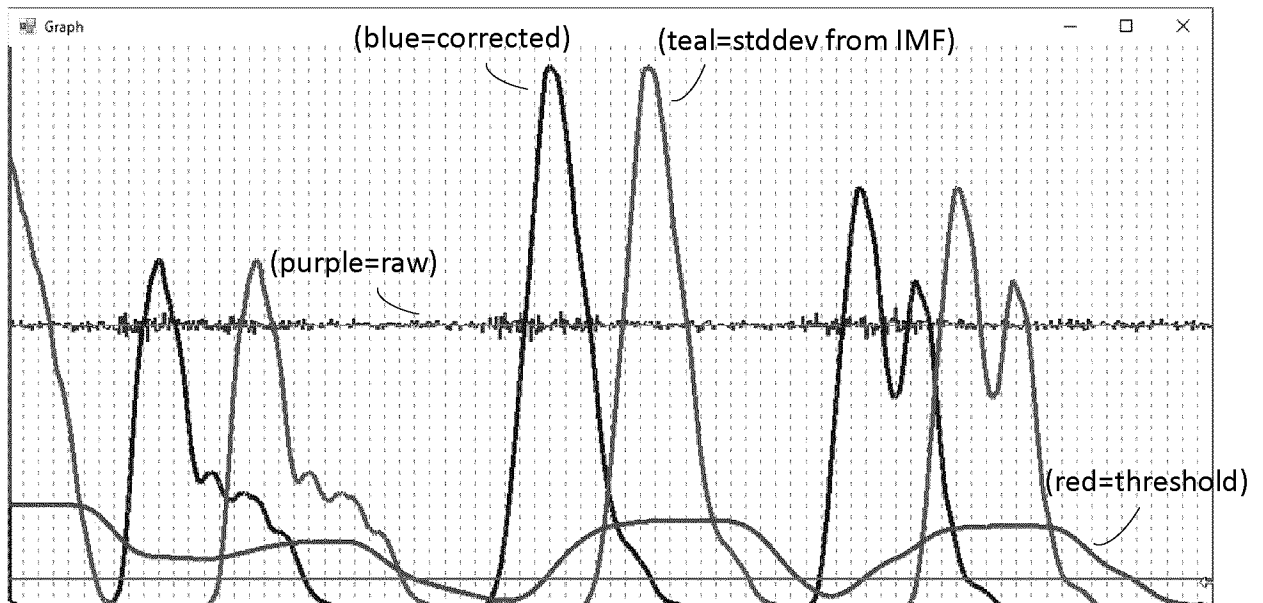
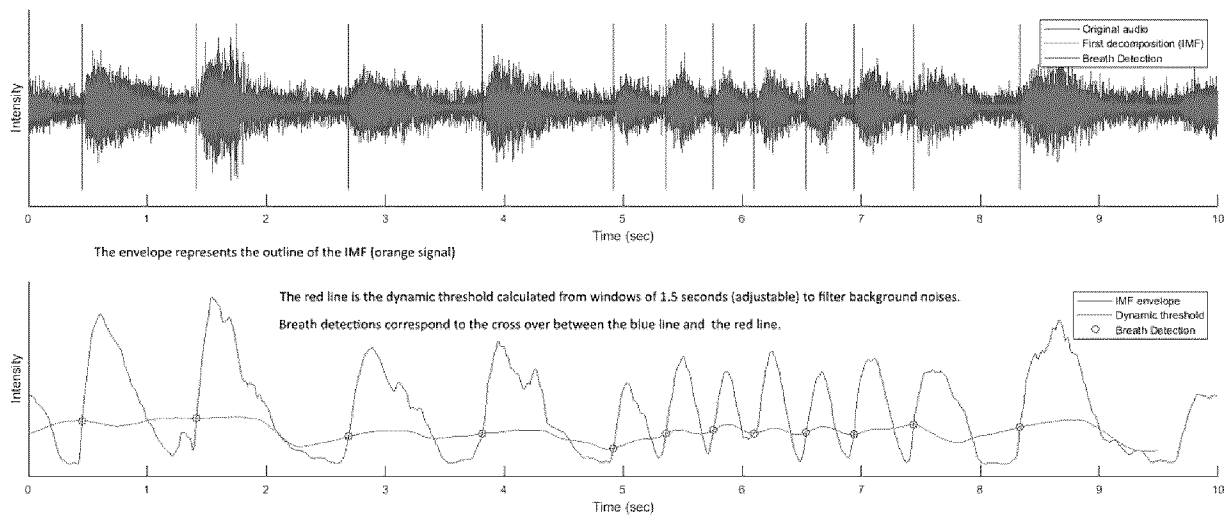


FIGURE 4

4/8

**FIGURE 5****FIGURE 6**

**FIGURE 7****FIGURE 8**

Initialize:

AbsoluteOffset = 0;

Executed for each audio recording:

Foreach AudioInputSample

```

NewSample = AudioInput // 1600 doubles at 16 000 hz
NewImf = EMD(NewSample) // Return IMF array, same size as new sample size
WindowIMF = [OldIMF + NewIMF]
WindowStdDev = [...] // SAMPLESIZE
WindowThreshold = [...] // SAMPLESIZE

For (j = 0; j < SAMPLESIZE; j++)
    stddev = AvgStdDevSimplified(WindowIMF, j, SampleSize) // Modified version of standard
    deviation, more detail below
    WindowStdDev[j] = stddev;
    newThresholdValue = MovingAvg.Add(stddev * ThresholdMultiplier);
    WindowThreshold[j] = newThresholdValue; // We smooth on 1.5s worth of record (15 samples,
    24000 doubles)
    // Beyond this point we omit signal alignment
    diff = (stddev - newThresholdValue) < 0 ? -1 : 1;
    if (diff != lastDiff) // Intersection between signal detected
        // Intersection detected!
        NewIntersection = [AbsoluteOffset, stddev]

        IF (stdDevAdd > 0) // If the addition is positive, then we have a breath, if it's negative, we
        have a silence
            // New breath detected !
            // Filter by intersection width (in seconds)
            // Check how much the stddev peak go relative to threshold to discriminate

        ELSE
            // SILENCE detected

        LastIntersection = newIntersection
        stdDevAdd = 0 // Reset, calculation need to be done until we get another intersection
        lastDiff = diff
        stdDevAdd += (stddev - newThresholdValue);
    AbsoluteOffset++;
    OldIMF = NewIMF
  
```

AvgStdDevSimplified:

```

static double avgStdDev_Simplified(double[] buffer, int start, int count)
{
    double output = 0;
    for (int i = start; i < start + count; i++)
    {
        double p1 = (buffer[i] - 0);
        output += p1 * p1;
    }
    return output;
}
  
```

FIGURE 9

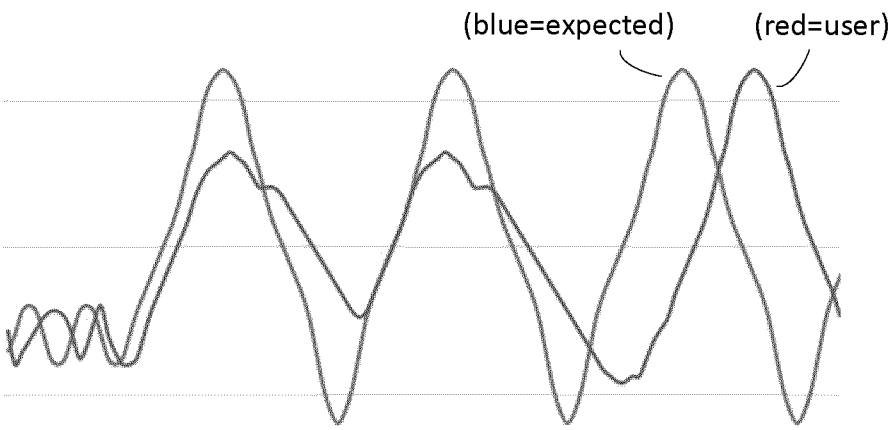


FIGURE 10

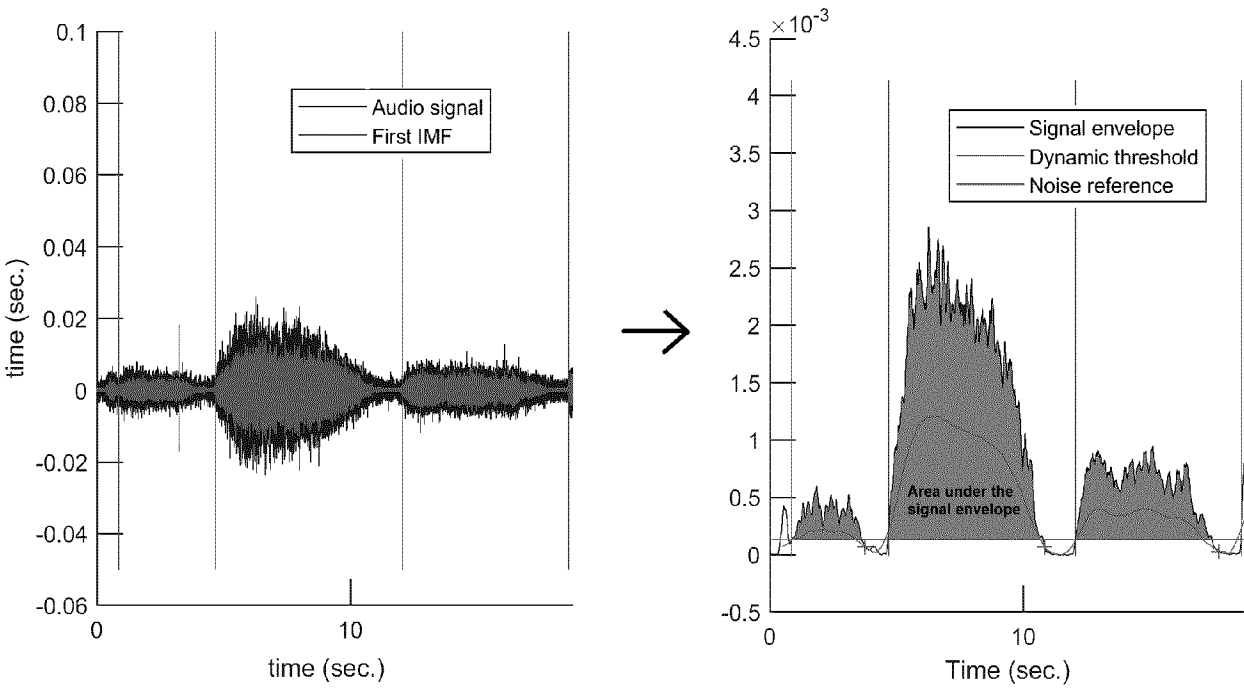


FIGURE 11

8/8

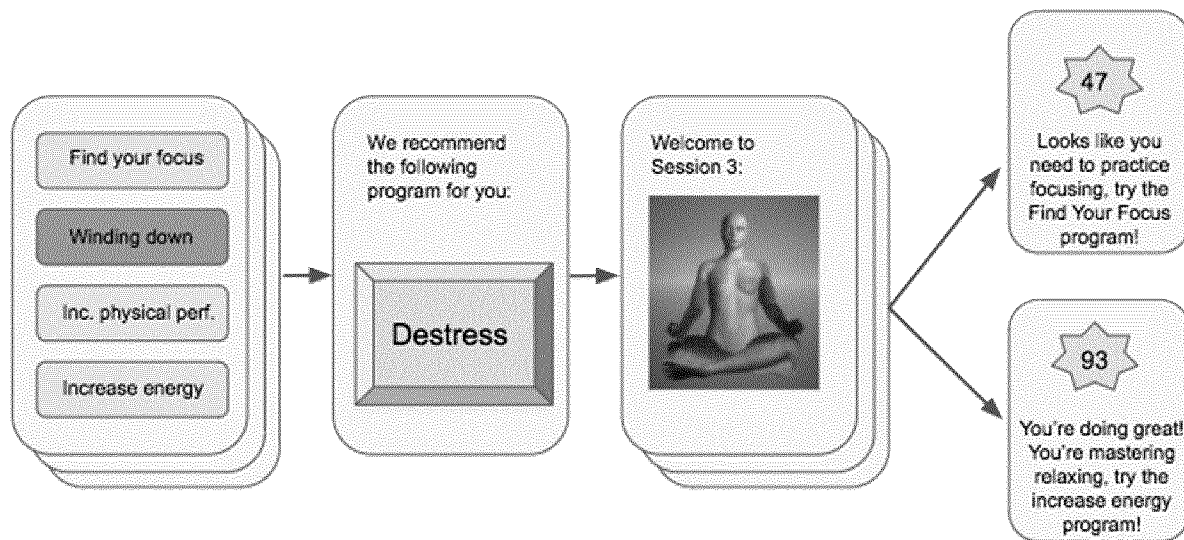


FIGURE 12

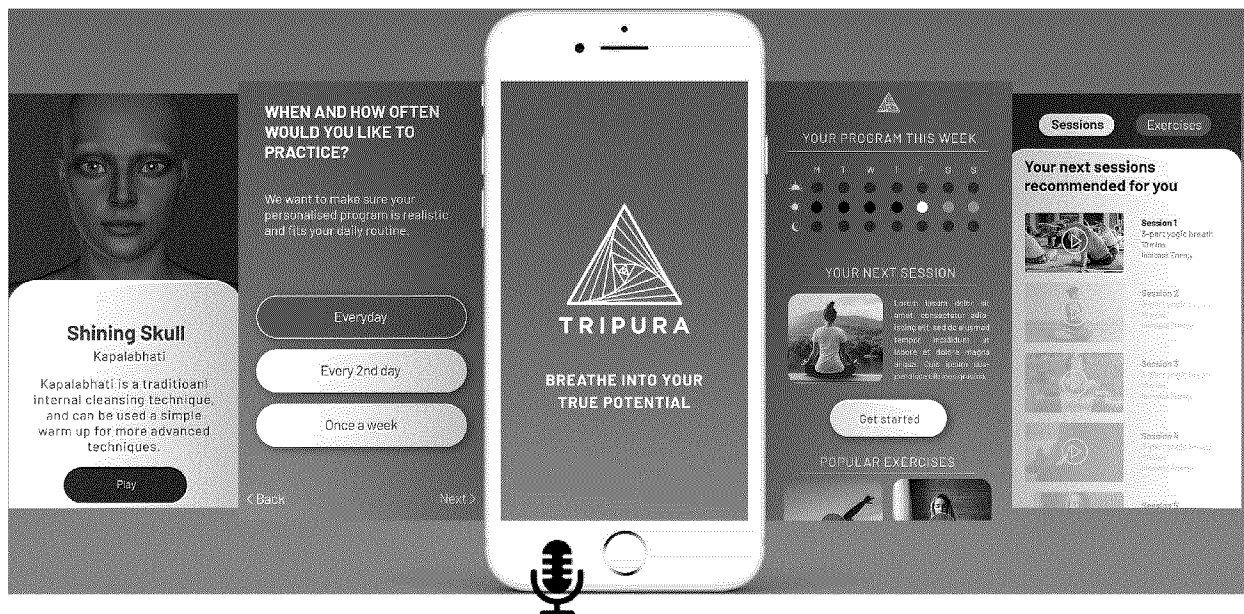


FIGURE 13

INTERNATIONAL SEARCH REPORT

International application No.

PCT/CA2020/050589

A. CLASSIFICATION OF SUBJECT MATTER

IPC: **G16H 20/30** (2018.01), **A61B 5/08** (2006.01), **A63B 23/18** (2006.01), **G10L 17/26** (2013.01),
G10L 25/03 (2013.01), **G10L 25/66** (2013.01)

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

G16H 20/30 (2018.01), A61B 5/08 (2006.01), A63B 23/18 (2006.01), G10L 17/26 (2013.01), G10L 25/03 (2013.01), G10L 25/66 (2013.01)

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic database(s) consulted during the international search (name of database(s) and, where practicable, search terms used)

Databases: WEST, Google Patents, Orbit, Canadian Patent Database

keywords: breathing, exercise, raw, data, microphone, library, game score, smartphone, app, efficiency compliance, match, pattern, timing event, biofeedback, audio

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	US 7 850 619 B2 (GAVISH et al.) 14 December 2010 (14-12-2010) see column 4, lines 19-35 and lines 58-67, column 7, lines 48-52, column 15, lines 55-67, column 16, lines 26-56, column 16, line 65 – column 17, line 9, column 18, lines 50-56, and column 19, lines 4-40	1-22
Y	US 2013/0331662 A1 (STOIAN et al.) 12 December 2013 (12-12-2013) see paragraph [0014]	1-22
Y	LANGE et al., "Breath: A game to motivate the compliance of postoperative breathing exercises", Virtual Rehabilitation International Conference, Haifa Israel, 29 June 2009 (29-06- 2009), .pp. 94-97, see The use of games for rehabilitation and C. The game concept	8, 9, and 20

☒ Further documents are listed in the continuation of Box C.

☐ See patent family annex.

* "A" "D" "E" "L" "O" "P"	Special categories of cited documents: document defining the general state of the art which is not considered to be of particular relevance document cited by the applicant in the international application earlier application or patent but published on or after the international filing date document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified) document referring to an oral disclosure, use, exhibition or other means document published prior to the international filing date but later than the priority date claimed	"T" "X" "Y" "&"	later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art document member of the same patent family
---	---	--------------------------	---

Date of the actual completion of the international search
24 July 2020 (24-07-2020)

Date of mailing of the international search report
29 July 2020 (29-07-2020)

Name and mailing address of the ISA/CA
Canadian Intellectual Property Office
Place du Portage I, C114 - 1st Floor, Box PCT
50 Victoria Street
Gatineau, Quebec K1A 0C9
Facsimile No.: 819-953-2476

Authorized officer

Kristina Bodnar (819) 635-3284

INTERNATIONAL SEARCH REPORT

International application No.

PCT/CA2020/050589

C (Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT		
Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	CHITTARO et al., " <i>Evaluating mobile apps for breathing training: The effectiveness of visualization</i> ". Computers in Human Behaviour, November 2014 (11-2014), Vol. 40, pp. 56-63, Retrieved from the Internet: < http://dx.doi.org/10.1016/j.chb.2014.07.049 > see whole document	
A	US 6 436 053 B1 (KNAPP, II et al.) 20 August 2002 (20-08-2002) see whole document	

INTERNATIONAL SEARCH REPORT
Information on patent family members

International application No.

PCT/CA2020/050589

Patent Document Cited in Search Report	Publication Date	Patent Family Member(s)	Publication Date
US7850619B2	14 December 2010 (14-12-2010)	US2008319333A1	25 December 2008 (25-12-2008)
		AU2005264165A1	26 January 2006 (26-01-2006)
		CA2574642A1	26 January 2006 (26-01-2006)
		CN101128150A	20 February 2008 (20-02-2008)
		EP1804649A2	11 July 2007 (11-07-2007)
		IL180866D0	31 October 2007 (31-10-2007)
		JP2008507316A	13 March 2008 (13-03-2008)
		KR20070048201A	08 May 2007 (08-05-2007)
		US2009118631A1	07 May 2009 (07-05-2009)
		US8485982B2	16 July 2013 (16-07-2013)
		US2013289431A1	31 October 2013 (31-10-2013)
		US9642557B2	09 May 2017 (09-05-2017)
		WO2006008745A2	26 January 2006 (26-01-2006)
US2013331662A1	12 December 2013 (12-12-2013)	WO2013185041A1	12 December 2013 (12-12-2013)
US6436053B1	20 August 2002 (20-08-2002)	CA2304541A1	08 April 1999 (08-04-1999)
		CA2381817A1	22 March 2001 (22-03-2001)
		EP1019147A1	19 July 2000 (19-07-2000)
		EP1215997A1	26 June 2002 (26-06-2002)
		JP2001518330A	16 October 2001 (16-10-2001)
		JP2003509099A	11 March 2003 (11-03-2003)
		US6106481A	22 August 2000 (22-08-2000)
		US2004015093A1	22 January 2004 (22-01-2004)
		US6740046B2	25 May 2004 (25-05-2004)
		WO0119245A1	22 March 2001 (22-03-2001)
		WO9916506A1	08 April 1999 (08-04-1999)