DESCRIPTION OF SLEEPWISE®

Respiratory Movement detection

SleepWise respiratory movement detector, continuously receives as input a plurality of images, I_1, I_2, …, I_n, and produces as output a binary array, $B(i, j)$, of one's and zero's where one's indicate pixel locations (i, j) at which respiratory motion has been detected.

Respiratory movement detector includes three phases; namely, (i) an image integrator that integrates a number, n, of live images recorded by a video recorder, (ii) a frame comparator that compares pixel values between images, and (iii) a noise filter that removes noise captured in the video recorder. Operating conditions of motion analyzer are such that the level of noise may be higher than the level of movement to be detected, especially in low light surroundings. Since motion analyzer is required to detect subtle movement, a challenge of the system is to appropriately filter the noise so as to maximize motion detection intelligence.

Typically, pixel values are specified by a rectangular array of integer or floating point data for one or more color channels. Familiar color systems include RGB red-green-blue color channels, CMYK cyan-magenta-yellow-black color channels and YUV luminance-chrominance color channels. For the present analysis, noise for color channel data is modeled as being Gaussian additive; i.e., if $I(i, j)$ denotes the true color data at pixel location (i, j) for a color channel, and if $G(i, j)$ denotes the color value measured by a video recorder, then

$$G(i,j) = I(i,j) + \varepsilon(i,j), \text{ where } \varepsilon(i,j) \sim \mathrm{N}(\mu, \sigma^2), \tag{1}$$

with mean μ, which is assumed to be zero, $\mu = 0$, and variance σ^2. Preferably, the values $I(i, j)$ are luminance values.

Image integrator receives as input a time series of n images, with pixel data denoted $G_1(i, j)$, $G_2(i, j)$, …, $G_n(i, j)$, and produces as output an integrated image $I(i, j)$. Preferably, image integrator reduces the noise level indicated in Equation (1) by averaging. Thus if $I(i, j)$ denotes the color data at pixel location (i, j) after integrating the n images, then the noise level can be reduced by defining:

$$I(i,j) = \frac{1}{n}\sum_{k=1}^{n} G_k(i,j). \tag{2}$$

As each additional image $G_{n+1}(i, j)$ is integrated within image integrator, the averaged pixel values are accordingly incremented dynamically as follows:

$$I(i,j) \leftarrow I(i,j) + \frac{G_{n+1}(i,j) - G_1(i,j)}{n}. \qquad (3)$$

For the present invention, an approximation to Equation (3) is used instead; namely,

$$I(i,j) \leftarrow I(i,j) + \frac{G_{n+1}(i,j) - I(i,j)}{n}, \qquad (4)$$

where *I(i, j)* has been used instead of $G_1(i, j)$. The advantage of Equation (4) over Equation (3) is that use of Equation (4) does not require maintaining storage of the raw image data $G_1(i, j)$, $G_2(i, j)$, …, $G_n(i, j)$ over a history of *n* images.

An advantage of averaging image data, as in Equation (2) above, is the elimination of noise. However, a disadvantage of averaging is that it tends to eliminate subtle movements, and especially periodic movement, making it hard to derive estimates of motion by comparing two images close in time. Thus in order to compensate for averaging, the present invention compares two images that are separated in time by a two seconds. In turn, this requires that a circular storage buffer of integrated images *I(i, j)* is maintained over a corresponding time span. For a video recording frame rate of, say, 15 frames per second, this corresponds to a circular buffer of approximately 30 images.

Image comparator receives as input the integrated images *I(i, j)* generated by image integrator , and produces as output a rectangular array, *Δ(i, j),* of binary values (one's and zero's) that correspond to pixel color value differences. Image comparator determines which portions of the images are moving, and operates by comparing two integrated images that are approximately 2 seconds apart in time. Preferably, image comparator uses differential changes instead of absolute changes, in order to avoid false movement detection when global lighting conditions change.

Denote by *IA(i, j)* and *IB(i, j)* two integrated images that are approximately two seconds apart in time, and that are being compared in order to extract motion information. Absolute differences such as $|IA(i,j) - IB(i,j)|$ are generally biased in the presence of a change in global lighting conditions. To avoid such a bias, image comparator preferably uses differential changes of the form:

$$\Delta(i,j) = \left|IA(i,j) - IA(i-\delta_1, j-\delta_2)\right| - \left|IB(i,j) - IB(i-\delta_1, j-\delta_2)\right|. \qquad (5)$$

Equation (5) incorporates both a spatial difference in a gradient direction (δ_1, δ_2), and a temporal difference over an approximate 2 seconds time frame. It is noted that a spatial difference generally eliminates global biases. Preferably, image comparator uses a sum of several such terms (5) over several different gradient directions.

After computing the differences $\Delta(i, j)$ at each pixel location (i,j), image comparator preferably uses a threshold value to replace $\Delta(i, j)$ with 1 for values of Δ greater than or equal to the threshold value, and to replace $\Delta(i, j)$ with 0 for value of Δ less than the threshold value. As such the output of image comparator is a binary array, $B(i, j)$, with values $B=0$ or $B=1$ at each pixel location (i, j).

The output of image comparator is passed to noise filter for applying active noise filters. Noise filter receives as input the binary array representing pixel color value differences produced by image comparator, and produces as output a corresponding noise-filtered binary array. Operation of noise filter is based on the premises that (i) motion generally shows up in multiple consecutive image differences, and not just in a single image difference; and (ii) motion generally shows up in a cluster of pixels, and not just in a single isolated pixel. Accordingly, noise filter modifies the binary array $B(i,j)$ by zeroing out values $B(i, j)=1$ unless those values of one's have persisted throughout some number, m, of consecutive comparison arrays B over time; and (ii) erosion is applied to the thus modified array $B(i, j)$ so as to zero out values of $B(i,j)=1$ at isolated pixels locations (i, j).

The binary array $B(i, j)$ output by noise filter identifies motion within the image; i.e., the pixel locations where $B(i, j)=1$ correspond to locations where motion is detected.

<u>Respiratory Event Detection</u>

Respiratory event detector receives as input the binary array, $B(i, j)$, of one's and zero's produced as output by high sensitivity motion analyzer and produces as output a movement signal, an airflow signal and a set of zero or more detected respiratory events for a period of time being analyzed.

As previously discussed, the result of respiratory movement detector is stored in binary array $B(i, j)$. In one embodiment, the resolution of the array B is equal to the resolution of the received image. Thus, each pixel in a received image corresponds to one position (i, j) in $B(i, j)$. The binary value of each position, or pixel, (i, j), in $B(i, j)$ indicates if the system has detected movement in that pixel. If movement has been detected the value of position (i,j) in $B(i, j)$ is one (1). If no movement is detected in that position the value is set to zero (0).

A movement signal generator uses stored arrays, $B(i, j)$, to generate a movement signal, $M(t)$, where $M(t)$ is a function that generates a single value for each received image that represents the total motion in the image. First, signal generator calculates a raw movement signal, $RM(t)$, by determining the number of nonzero pixels in $B(i, j)$. In other words, $RM(t)$ is the number of pixels in the array $B(i, j)$ where the system has detected movement. $B(i, j)$ in this case refers to the motion array for time t.

The movement signal generator then applies a smoothing filter, to further reduce noise, to $RM(t)$ to produce movement signal $M(t)$. In a preferred embodiment, the smoothing filter averages n adjacent values of $M(t)$. The value of n is based on the sample rate, i.e. the number of images received per second. For example, ten samples received during a one second interval are averaged. Thus, $M(t)$ is obtained by:

$$M(t) = \frac{1}{n}\sum_{k=1}^{n} RM\ (t - rnd(n/2) + k) \quad (7)$$

For each t, this equation yields a value for *M(t)* that is an average across the n adjacent values of *RM(t)* centered at t. The function rnd(n/2) rounds the value n/2 up to the nearest integer in the case that n is an odd integer.

Next, an envelope detector calculates a movement envelope for the movement signal, *M(t)*. Envelope detector receives as input the movement signal *M(t)* and produces as output two signals, or series, an upper envelope $M_{upper}(t)$ and a lower envelope, $M_{lower}(t)$ as follows:

$M_{upper}(t)$ is the maximum value that *M(t)* takes on during a time interval, I, centered at time t.

$M_{lower}(t)$ is the minimum value that *M(t)* takes on during a time interval, I, centered at time t.

A preferred value for time interval I is 1 second, but longer or shorter values can be used. Thus, the upper and lower movement envelopes are line segments that, taken together, enclose the values of *M(t)*.

An airflow signal generator creates an airflow signal, *S(t)*, by adding the lower and upper value of the movement envelope at each point. Thus:

$$S(t) = M_{upper}(t) + M_{lower}(t) \quad (8)$$

Air flow signal, *S(t),* represents the amount of air flowing into the lungs of the subject over time. When the subject breathes deeply the corresponding movement and hence the values of the airflow signal increase during the period of deep breathing ; when the subject breathes shallowly, or not at all, his/her movement decreases and hence the values of the airflow signal decrease during the corresponding period.

A respiratory event classifier receives airflow signal *S(t)* and analyzes it to detect respiratory events.

Respiratory events, as identified by respiratory event classifier, occur when a significant decrease in the airflow signal is followed by a significant increase in the airflow signal. The first step is to detect significant decreases in the air flow signal. This is performed, by identifying the intervals [t0, t1] where the derivative of the airflow signal, *S(t)* is uniformly negative, in other words interval [t0, t1] satisfies:

$$\forall\ t,\ t0<t<t1,\ (S(t)-S(t+\delta)) < 0 \quad (9)$$

Here *ä* is the sample interval; in other words, if *S(t)* is the value of airflow signal *S(t)* at time *t*, then *S(t+ δ)* is the value of the airflow signal that corresponds to the next received image.

At step significant increases in the air flow signal are detected. This is performed, by identifying the intervals [t0, t1] where the derivative of the airflow signal is uniformly positive, in other words interval [t0, t1] satisfies:

$$\forall\ t,\ t0<t<t1,\ (S(t)-S(t+\delta)) > 0 \qquad (10)$$

Here again *ä* is the sample interval such that if *S(t)* is the value of airflow signal *S(t)* at time *t*, then *S(t+ δ)* is the value of the airflow signal corresponding to the next received image.

Each case where a significant decrease in the airflow signal, is followed by a significant increase in the airflow signal, is considered a candidate respiratory event. A candidate event then is a time interval that begins when the derivative of an airflow signal becomes negative until the derivative of the airflow signal reverses and becomes positive.

However, not all the candidate respiratory events qualify as true respiratory events. From a clinical point of view, a decrease in the airflow is not considered a respiratory event if it does not have a significant impact on the saturation of oxygen in the blood. Thus, empirically derived tests are applied to the candidate events in order to discard events that are determined to be insignificant. The remaining candidate events are then deemed to be true respiratory events. The empirical tests are: duration of the decrease/increase in the airflow signal and the derivative of the airflow signal. Additional or different tests may also be applied to the candidate respiratory events, either to discard false events or to identify true respiratory events, like, for example, contextual information, it is, if a respiratory event is isolated or, as usual, forms part of a series of respiratory events.

Duration of the decrease. If the interval *[t0, t1]* when a decrease in the air flow signal occurred, lasts less than a threshold period of time, referred to as the Decrease Duration Threshold, the interval will be discarded because the corresponding decrease in the air flow is too brief to significantly influence the saturation of the oxygen in the blood.

Derivative of the airflow signal. The average derivative of the airflow signal, *S(t)*, during a candidate interval *[t0, t1]* must initially, during the period of decreasing respiration, exceed a negative threshold, referred to as the Negative Derivative Threshold. In the next phase, when respiration increases, the average derivative must exceed a Positive Derivative Threshold. The derivative threshold test is calculated using an approximation of the average derivative of *S(t)* over the interval *[t0, t1]* as follows:

During the initial period of decreasing airflow:

$$\frac{\max[S'(t)]-\min[S'(t)]}{dur[t0,t1]} \leq Negative\ Derivative\ Threshold \qquad (11)$$

And, during the following period of increasing airflow:

$$\frac{\max[S'(t)] - \min[S'(t)]}{dur[t0,t1]} \geq Positive\ Derivative\ Threshold \quad (12)$$

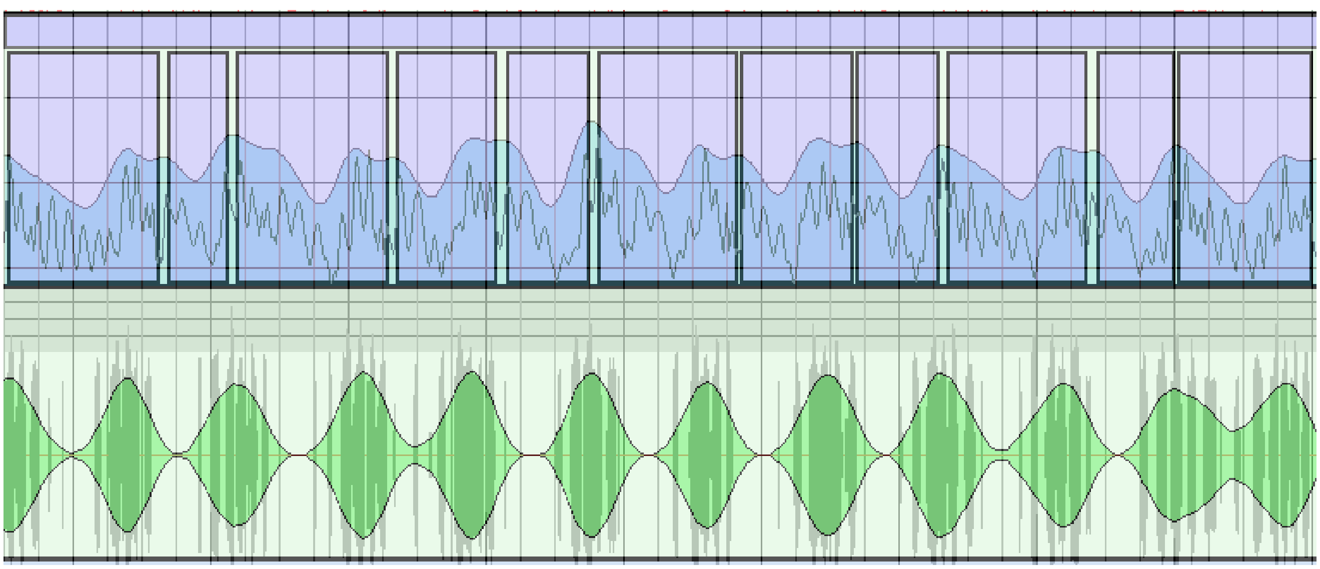
where

max[S'(t)] is the maximum value of the derivative of *S(t)* over the interval *[t0, t1]*,

min[S'(t)] is the minimum value of the derivative of *S(t)* over the interval [t0, t1], and

dur[t0, t1] is the duration of the time interval *[t0, t1]*, typically measured in seconds.

The values of the two thresholds, Decrease Duration Threshold and Derivative Threshold, are established using clinical test results obtained using polysomnography. Essentially, the events detected using the subject invention are compared with those obtained using a polysomnography apparatus in a clinical environment such as a sleep center or a hospital. The, using the subject invention to perform a comparable analysis the various thresholds are systematically varied to yield the best comparative results. This is performed over a number of sleep sessions involving different subjects to further tune the threshold values.



Example of a series of apneas. Each apnea is framed in a lilac box. Observe the attenuation on the inferred airflow (blue area on the respiratory movement signal also in dark blue) during the apnea and increase movement detected during the end of the apnea. Note also that the end of the apnea matches with snoring (audio signal in green).

<u>State of sleep/awake Detection.</u>

State Detector module is used to infer the subject state (sleep/awake). This module include (i)RuggedMovement extractor, Rugged Movement event detector(ii) and state classifier(iii). This module produces as output a set of periods of time indicating if the patient is sleep or awake in each period.

Rugged Movement signal extractor continuously receives as input a plurality of images, I_1, I_2… I_n, of resolution cx pixels wide and cy pixels height. Denote by $I_n(i, j)$ and $I_{n+2}(i, j)$ images that are approximately two seconds apart in time, and that are being compared in order to extract motion information. The motion information consist in calculate a global difference between both images according with the equation:

$$E_n = \frac{\sum_{i=0}^{cx} \sum_{j=0}^{cy} | I_n(i, j) - I_{n+2(}i, j) |}{cx * cy}$$

The signal Rugged Movement RM(t) contains the value of the E_n for each frame processed in a given time and represents how significant is difference between two images that are two seconds apart of time. Observe that this method for extracting movement is less sophisticate that the respiratory movement detector. In fact, this module is used just to detect when the subject is making a big movement.

The signal RM(t) in a given time will have a low value if the two compared images are very similar and will have a high value if both images contains big differences. Generally when the values are low means that the subject is relaxed and just breathes. Otherwise the subject is changing his body position or eventually is moving a limb.

A Rugged Movement events, occur when the values signal RM(t) exceeds a threshold. The threshold is fixed as the average of the RM(t) values during all the recording time.

The Rugged Movement event detector identifies, localizes in time and mark the duration of each Rugged Movement event during recording.

The state classifier uses the information provided by Rugged Movement event detector to determine if the patient is sleep or awake.

Following a similar principles that uses the actigraphy for sleep state classification, the state classifier find look for the events with duration superior to 10 seconds. This events will determine the beginning of awake periods. The duration of the awake period is extended as the classifier finds events within a period of time from the beginning of the awake period or from the last Rugged Movement event found. If no event is found within the fixed period of time the classifier will mark the end of the awake period.

Once again, the values of the several thresholds involved in the classification of the patient state are established using clinical test results obtained using polysomnography.