

PROCESSING TIME-SERIES DATA AND RELATED SYSTEMS AND METHODS

Technical Field

5

[0001] The present invention relates, in general terms, to a system for processing time-series data such as electroencephalogram (EEG) data and photoplethysmogram (PPG) data.

10 Background

[0002] Many different devices measure different kinds of signals and transmit using different communication channels. In many instances, these signals relate to a common entity – e.g. blood pressure and heart rate signals. As a result, hybrid or mixed model studies based on multiple source data streams are emerging.

[0003] One major hurdle with analysing information across multiple sources is the variation in sampling rate. Delays across communication channels differ, resulting in data from multiple sources appearing out of synch even where they have the same sampling rate. This is also problematic where data is buffered, since the system can assume that the timestamps for the data are very close, followed by a larger gap when the buffer is refilling.

[0004] Synchronization between data sources is a challenge, as mismatched data makes decisions unreliable.

[0005] It would be desirable to overcome or ameliorate at least one of the above-described problems, or at least to provide a useful alternative.

Summary

30

[0006] Disclosed herein is a method to synchronize data from multiple sources. It is suitable for both offline data analysis and model training

and for online data processing applications.

[0007] The present invention provides a system for producing synchronised time-series data, the system comprising:

one or more processor(s), a memory comprising program code executable by the processor(s):

receive a first time-series data from a first measurement device and a second time-series data from a second measurement device, wherein the first and second time-series data comprise a series of data points and an arrival timestamp for each data point;

process the received first and second time-series data using a regression model to obtain processed first and second time-series data respectively, wherein the regression model performs regression on the received arrival timestamps of the first and second time-series data to generate the processed first and second time-series data comprising synchronised timestamps, thereby synchronising the processed first and second time-series data with respect to each other.

[0008] A method for producing synchronised time-series data, the method comprising:

receiving a first time-series data from a first measurement device and a second time-series data from a second measurement device, wherein the first and second time-series data comprise a series of data points and associated arrival timestamps;

processing the received first and second time-series data using a regression model to obtain processed first and second time-series data respectively, wherein the regression model performs regression on the received arrival timestamps of the first and second time-series data to generate the processed first and second time-series data comprising synchronised timestamps, thereby synchronising the processed first and second time-series data with respect to each other.

Brief description of the drawings

- [0009] Embodiments of the present invention will now be described, by way of non-limiting example, with reference to the drawings in which:
- 5 [0010] Figure 1 illustrates a method for producing synchronised time-series data, along with a linear regression algorithm implemented in that method, in accordance with present teachings;
- [0011] Figure 2 shows an example of input, received and linear-regression-based corrected data points;
- 10 [0012] Figures 3(a) to 3(d) show a delay simulation of two sine wave data in which Figure 3(a) is the simulated delay generated by a random model, with mean delay of 1.5 seconds, blue for signal 1 and red for signal 2, image Figure 3 (b) shows the simulated received data with generated delay values, image Figure 3 (c) shows received data processed with our linear regression algorithm, and image Figure 3 (d) plots the difference
- 15 between the two signals, blue for received data with S2 interpolate with S1 timestamps, without any processing, red for Linear Regression processed data, with S2 interpolated with S1 timestamps;
- [0013] Figure 4 shows the mean squared error (MSE) with randomize index changes from 0.1 to 1 and system running over 100 cycles;
- 20 [0014] Figure 5 shows data points generated by a Bluetooth device measuring both EEG and PPG with different sampling rates;
- [0015] Figure 6 shows the same data points as those shown in Figure 5, but after linear regression-based synchronisation in accordance with
- 25 present teachings; and
- [0016] Figure 7 shows a system for implementing the method of Figure 1.

Detailed description

- 30 [0017] Data synchronisation techniques used herein employ regression techniques for merging timestamps for data from multiple sources. In general, these techniques leverage linear regression. Typically, linear

regression algorithms are designed for one source of data. The present teachings extend linear regression to multiple data source synchronization.

[0018] Figure 1, illustrates a method 100 for producing synchronised time-series data. Synchronisation is based on the understanding that, in general, a measurement device will generate samples at a fixed sampling rate. The term "samples" will be interchangeably used with "data points".

[0019] Despite the sampling rates being fixed, data arrives after variable lengths of time. This variation is largely due to delays in processing and transmission operations performed on or with the data. As a result, the sequence of data from multiple data sources can be disturb relative to each other – e.g. a first data point that was generated by one device slightly before a second data point from another device may in fact be received after the second data point. So, synchronisation as used herein results in the order at time of the arrival of data points being preserved, despite different delays or communication latencies.

[0020] The method 100 involves receiving the time series data (step 102) and processing it to obtain processed time series data (step 104). Given the method 100 synchronises data across multiple devices, step 102 involves receiving first and second time-series data from first and second measurement devices, respectively. The same teachings can be extended to three or more devices either by synchronising using the timestamps of one of the multiple devices, or synchronising using the system clock. The first and second time-series data comprise a series of data points and an arrival timestamp for each data point.

[0021] Step 104 then involves processing the received first and second time-series data using a data processing model. This output is processed first and second time-series data. Where the timestamps of one time series data (for illustration purposes, the first time series data) are used to determine timestamps for the other time series data, then the first time series data may be the same as the processed first time series data – i.e. it remains unchanged.

[0022] The data processing model is a regression model that performs regression on the received arrival timestamps of the first and second time-series data. The regression model generates the processed first and second time-series data such that they have synchronised timestamps.
5 As a result, the regression model synchronises the processed first and second time-series data with respect to each other.

[0023] For multiple data sources, there are two types of synchronization – master source synchronization and system paced synchronisation. In all cases, interpolation is intended to reproduce the original signal using
10 a common reference clock. This is because multi-signal or multi-source data processing generally assumes that data points are simultaneously sampled. In master source synchronisation, one data source determines the pace of the output data samples for all other data sources. All other data sources need to do sample interpolation if necessary. By
15 synchronising all time-series data with respect to one time-series data (data from the master data source), there is no need to interpolate timestamps for the master data source. This can be useful if a particular source has a more accurate clock than the system clock, or if subsequent processing steps make use of or refer to the sample rate or clock or a
20 particular source (the master source). System paced synchronisation uses a fixed system time step, e.g. generated by the system clock, to output data samples. There is no master data source, and all data sources may need to have timestamps interpolated. Advantageously, however, the output-sampling rate is fixed.

25 [0024] Each form of synchronisation takes the same data inputs, and the data are asynchronously supplied, since every data source has its own sampling rate and delays on the time that data takes to be received by the system running method 100.

[0025] Step 104 may comprise passing each time-series data to a
30 corresponding, or separate, linear regression instance for producing the processed timestamps. In this instance, each linear regression instance runs on a processor associated with memory. The memory stores a queue

of the time-series data for later synchronisation. This can be used to match the data with the processed (calculated) timestamps and/or to maintain a sliding window of data that is buffered for synchronisation with data points that are subsequently received.

5 [0026] According to master source data synchronisation, the output data is set based on the master data source. In this case, a number of samples are acquired for one data source (the master data source), the total number of samples is determined and corresponding timestamps are calculated. This can be done by interpolating over a known time window.
10 For example, if 60 data points are received in a one second window, a 60Hz sampling rate will be assumed even if some samples are yet to be received, and the sequence of data points are allocated timestamps of successive intervals of $1/60^{\text{th}}$ of a second.

[0027] The data points, ordered by processed timestamps, form a list.
15 That list is used as a parameter to check all other data sources for to determine a number of data points from those other data sources that are available in the time period covered by the list – in the 60Hz examples, with 60 samples, the data points from the other data sources will be those that occur in the one second time period over which the data
20 points in the list were captured.

[0028] For each timepoint available data points are collected by calling the function 106 set out in Figure 1. If there are no data points for a data source, the function returns null or 0. Otherwise, it gets one sample and its timestamp from the master data source, then use the timestamp as
25 parameter to get data sample from all other data sources where data interpolation is needed. Then append all data samples together and return.

[0029] In the case of system paced synchronisation, the output sampling rate corresponds to the system clock and thus all data points are
30 synchronised with respect to the system clock. A variable, next-ts, is set to store the timestamp of the next output sample: next-ts.

[0030] Again, the number of available data points is determined. The current timestamp is obtained according to the system clock, and the latest processed timestamps are obtained from the data sources. The next-ts is then checked: if it is empty (marked as NAN: not a number), it is set as the latest processed timestamp. The number of samples is then obtained for each data source over the period from the previous system clock generated timestamp to next-ts.

[0031] The algorithm 106 receives an input stream of time-series data y . y is then stored in a buffer. A count c is taken of the number of data points in y . If c is less than a current length of data used for regression, $xlen$, (i.e. current number of data points, such as the number of data points from the master data source in the list), then regression is performed over the count c for the time period covered by $xlen$. In this case De is used to evaluate if there is sufficient data for accurate regression model computation. Then timestamp for the first time entry for the data source $y1$ is set as the timestamp used for obtaining the single data point. If the count is greater than $xlen$ but less than the length of the buffer (set empirically), then the time step $Xstep$ for the subsequent sample will be set to be $c/xlen$.

Y_{sum} is given by:

$$Y_{sum} = \sum_{j=1:xstep:N} y_j$$

where j is the most recent sample in input stream y at timestep and N is the total number of timesteps over which regression takes place.

[0032] Linear regression process is applied to data arrival time – i.e. the time at which a data point is received at the system implementing method 100. There is an assumption that at the device site, the sampling rate is fixed. The variances in arrival time are caused by data processing and transmission time.

[0033] The system will thus receive data point x at time y as below:

$$X = [x_0, x_1, \dots, x_{n-1}]$$

$$Y = [y_0, y_1, \dots, y_{n-1}]$$

5

[0034] Using simple linear regression to derive a formula:

$$t_i = a * x_i + b$$

10 [0035] The following formulas can be used to calculate "a" and "b": (here i is the index of the sample in the buffer at every time step $xstep$)

$$a = \frac{n \sum_i (x_i y_i) - \sum_i x_i \sum_i y_i}{n \sum_i x_i^2 - (\sum_i x_i)^2}$$

$$b = \frac{\sum_i y_i \sum_i x_i^2 - \sum_i x_i \sum_i (x_i y_i)}{n \sum_i x_i^2 - (\sum_i x_i)^2}$$

15

[0036] A linear-regression function takes data sample arrival timestamp (the time at which the data was received) as input, and returns the processed timestamp. A data buffer is created to keep an historical record of a sliding window of data points for parameter updating.

20

[0037] To improve system-processing speed, regression is done with a step-size parameter, $xstep$ (the number of data points per regression parameter update – i.e. the number of data points before the regression model is recomputed, bearing in mind that regression over a continuous stream of data cannot be done in real time over the entire data set and the present methodology makes accurate regression by windowing the data stream every $xstep$ data points). This is to allow a configurable balance between the processing speed and accuracy of the results. At beginning, the step is 1. When the number of data sample is increased to big enough, the step is automatically increased by 1. This process continues until the data buffer, y -buffer, is full.

25

30

[0038] Figure 2 shows a simulated sine signal, x is time and y is amplitude. The sampling rate is 250 Hz. The original data is shown in thick solid line, which is a standard 2 Hz sine signal. Due to the transmission and buffering latency, the end side received data in thin solid line shows zigzag shapes when plotted against the time (axis X). Applying linear regression algorithm to the received timestamps, the processed data shown in dashed line has shape very close the original data, except a few data at the beginning not shown the smooth signal, due to too few data points contributing to the regression at beginning. After a very short while (around 100 milliseconds), the output becomes stable, and the processed data (orange) shows very similar shape to the original data, with a short delay in time.

[0039] To demonstrate the effect of algorithm 106, a data delay model was used to simulate data delay during data transmission process. The process is illustrated in Figure 3(a) to (d). A delay time was generated, in seconds, was generated segment by segment – e.g. for each data point – as reflected in Figure 3(a). This was achieved by setting an average delay time (for example 1.5 seconds), then applying a random time duration, generated by a random number generator, to increase and decrease the delay to a target length within a range controlled by an input variable. This is referred to as the randomization rate. The value of the randomization rate can be changed from 0 to 1. With this model, the algorithm can be evaluated as follows:

1. Create two sine wave signals of the same frequency (for example 5Hz), at different sampling rates, 520 Hz and 480 Hz respectively. The time sequence of T1 and data X1 are generated for signal 1, and time sequence T2 and data X2 are generated for signal 2.
2. The above-mentioned model is used to generate a delayed time sequence Trcv_1 for signal 1 and Trcv_2 for signal 2 (Figure 3(a)). The resulting signal plots X1 vs Trcv_1, X2 vs Trcv_2 are shown in Figure 3(b) to compare the differences.

3. The linear regression algorithm is then called on Trcv_1 to get tr1, Trcv_2 to get tr2. Plot X1 vs tr1 and X2 vs tr2 as shown in Figure 3(c). This shows close concordance between the original input sine waves and the sine waves after timestamp correction.

5 4. To determine and compare the errors cause by delays, both for received data and processed data, the differences are calculated between X1 and interpolated X2 at S1 time sequences (i.e. the timestamps corresponding to master data source, T1). As shown in Figure 3(d), the solid line curve shows the differences of X1 and X2 with Trcv_2 interpolated at Trcv_1, and plotted against time sequence Trcv_1, and the dashed line shows the differences between X1 and X2 with tr2 interpolated at tr1, plotted against time sequence tr1. MSE of X1 and interpolated X2 are calculated to evaluate the linear-regression algorithm.

15 [0040] To further investigate, Figure 4 shows the effect, over 100 cycles of operation, of changing the delay generation model randomization parameter from 0.1 to 1, with the MSE calculated for individual rates. The solid line shows the mean and standard derivation of the MSE for received data without linear regression and the dashed line shows the mean of stand derivation of the MSE for data points processed using the linear regression algorithm. While the error deteriorates (increases) as randomisation increases, the output nevertheless tracks the original, pre-delay signal, far more closely than in the absence of regression analysis.

20 Moreover, even at a randomization rate of 1, the data averages 0.5 error, which is understandable as it reflects that average between 0 and the maximum error of 1. The lower accuracy rate may be due to data points arriving so early, or late, when compared with neighbouring data points, that they are included in a time window that is earlier, or later, than the time window in which they would otherwise have been processed.

30 [0041] In some applications, the method 100 operates on electroencephalogram (EEG) and photoplethysmogram (PPG) data. In

some embodiments, a wireless EEG device captures a forehead EEG signal as well as PPG signal, sampling at different rates: 256Hz and 60Hz. Other than PPG and EEG, there are also accelerometer and gyroscope sensor data that can be captured, as well as temperature, battery level and other kinds of data. All data are sensed at different sampling rates, as reflected in Figure 5. All data are transferred to the system (e.g. that shown in Figure 7) through Bluetooth LE wireless communication protocols. The EEG and PPG data and battery level of the measuring device are captured. The EEG and PPG data facilitate real time display and processing.

[0042] Figure 5 shows the data points prior to regression being applied. It is obvious that there are gaps in the data being obtained in the PPG data channel 500, between consecutive packages. This makes the displayed PPG signal zig-zagged. Consequently, the PPG data cannot be used directly for analysis and application.

[0043] The same data points are shown in Figure 6, after linear regression has been applied for data synchronisation. Clearly, the PPG data indicated by reference numeral 600, and also the EEG data, are smoothened. It is expected that the smoothened data considerably more closely matches the original data, particularly when observing the section marked 502 in Figure 5, when compared with the same data points after correction as reflected at 602 in Figure 6.

[0044] Figure 7 shows a system implementing the method 100. The system 700 comprises at least one processor 702, memory 704 and a network interface 706 for communicating with measurement devices 708 over a wired or wireless network 710. The memory comprises program code or instructions that, when executed by processor(s) 702 implement the method 100. The program code 712 implements a time-series data processing module 714 that runs a regression model 716 to process received time series data. There may be a separate regression model for each time-series data input.

[0045] The system 700 may be any suitable device, such as a personal computer, smartphone or other device. Moreover, the separate program code modules and models may be implemented in any desired manner, such as in hardware or software. Also, while the system 700 has been
5 shown as a single device, it, or components thereof, may be distributed across multiple servers.

[0046] It will be appreciated that many further modifications and permutations of various aspects of the described embodiments are possible. Accordingly, the described aspects are intended to embrace all
10 such alterations, modifications, and variations that fall within the spirit and scope of the appended claims.

[0047] Throughout this specification and the claims which follow, unless the context requires otherwise, the word "comprise", and variations such as "comprises" and "comprising", will be understood to imply the inclusion
15 of a stated integer or step or group of integers or steps but not the exclusion of any other integer or step or group of integers or steps.

[0048] The reference in this specification to any prior publication (or information derived from it), or to any matter which is known, is not, and should not be taken as an acknowledgment or admission or any form of
20 suggestion that that prior publication (or information derived from it) or known matter forms part of the common general knowledge in the field of endeavour to which this specification relates.

25

Claims

1. A system for producing synchronised time-series data, the system comprising:
 - 5 one or more processor(s), a memory comprising program code executable by the processor(s):
receive a first time-series data from a first measurement device and a second time-series data from a second measurement device, wherein the first and second time-series data comprise a series of data points and an
10 arrival timestamp for each data point;
process the received first and second time-series data using a regression model to obtain processed first and second time-series data respectively, wherein the regression model performs regression on the received arrival timestamps of the first and second time-series data to generate the
15 processed first and second time-series data comprising synchronised timestamps, thereby synchronising the processed first and second time-series data with respect to each other.
2. The system of claim 1, wherein the regression model performs regression
20 based on a step-size parameter that defines a number of data points to be processed in a single iteration by the regression model to generate the respective processed time-series data.

3. The system of claim 1, wherein the regression model is a linear regression model executed independently for each received time-series data, the linear regression model defined by the formulae:

Received data points: $x = [0, 1, \dots, x_{n-1}]$

Arrival time: $t = [t_0, t_1, \dots, t_{n-1}]$

Synchronised timestamps: $t_i = a * x_i + b$

$$a = \frac{n \sum_i (x_i t_i) - \sum_i x_i \sum_i t_i}{n \sum_i x_i^2 - (\sum_i x_i)^2}$$

$$b = \frac{\sum_i t_i \sum_i x_i^2 - \sum_i x_i \sum_i (x_i t_i)}{n \sum_i x_i^2 - (\sum_i x_i)^2}$$

4. The system of any one of claims 1 to 3, wherein the first measurement device is an EEG (electroencephalogram) device and the first time series data is EEG data, and the second measurement device is a PPG (photoplethysmogram) device and the second time series data is PPG data.

5. The system of any one of claims 1 to 4, wherein the first time-series data is treated as a master time-series data source and the processed second time series data is generated at a frequency to match an arrival frequency of the master time-series data.

6. A method for producing synchronised time-series data, the method comprising:

receiving a first time-series data from a first measurement device and a second time-series data from a second measurement device, wherein the

first and second time-series data comprise a series of data points and associated arrival timestamps;

processing the received first and second time-series data using a regression model to obtain processed first and second time-series data respectively, wherein the regression model performs regression on the received arrival timestamps of the first and second time-series data to generate the processed first and second time-series data comprising synchronised timestamps, thereby synchronising the processed first and second time-series data with respect to each other.

7. The method of claim 6, wherein the regression model performs regression based on a step-size parameter that defines a number of data points to be processed in a single iteration by the regression model to generate the respective processed time-series data.

8. The method of claim 6, wherein the regression model is a linear regression defined by the formula:

Received data points: $x = [0, 1, \dots, x_{n-1}]$

Arrival time: $t = [t_0, t_1, \dots, t_{n-1}]$

Synchronised timestamps: $t_i = a * x_i + b$

$$a = \frac{n \sum_i (x_i t_i) - \sum_i x_i \sum_i t_i}{n \sum_i x_i^2 - (\sum_i x_i)^2}$$

$$b = \frac{\sum_i t_i \sum_i x_i^2 - \sum_i x_i \sum_i (x_i t_i)}{n \sum_i x_i^2 - (\sum_i x_i)^2}$$

9. The system of any one of claims 6 to 8, wherein the first measurement device is an EEG (electroencephalogram) device and the first time series data is EEG data, and the second measurement device is a PPG (photoplethysmogram) device and the second time series data is PPG data.

5

10. The system of any one of claims 6 to 9, wherein the first time-series data is treated as a master time-series data source and the processed second time series data is generated at a frequency to match an arrival frequency of the master time-series data.

10