

Gustav: Cross-device Cross-computer Synchronization of Sensory Signals

Kayhan Latifzadeh
Luis A. Leiva
name.surname@uni.lu
University of Luxembourg
Luxembourg

ABSTRACT

Temporal synchronization of behavioral and physiological signals collected through different devices (and sometimes through different computers) is a longstanding challenge in HCI, neuroscience, psychology, and related areas. Previous research has proposed to synchronize sensory signals using (1) dedicated hardware; (2) dedicated software; or (3) alignment algorithms. All these approaches are either vendor-locked, non-generalizable, or difficult to adopt in practice. We propose a simple but highly efficient alternative: instrument the stimulus presentation software by injecting supervisory event-related timestamps, followed by a post-processing step over the recorded log files. Armed with this information, we introduce Gustav, our approach to orchestrate the recording of sensory signals across devices and computers. Gustav ensures that all signals coincide exactly with the duration of each experiment condition, with millisecond precision. Gustav is publicly available as open source software.

CCS CONCEPTS

• **Human-centered computing** → **Laboratory experiments; Interaction devices**; HCI design and evaluation methods.

KEYWORDS

Orchestration; Synchronization; Behavioral/Physiological Sensing

ACM Reference Format:

Kayhan Latifzadeh and Luis A. Leiva. 2022. Gustav: Cross-device Cross-computer Synchronization of Sensory Signals. In *The Adjunct Publication of the 35th Annual ACM Symposium on User Interface Software and Technology (UIST '22 Adjunct)*, October 29–November 2, 2022, Bend, OR, USA. ACM, New York, NY, USA, 3 pages. <https://doi.org/10.1145/3526114.3558723>

1 INTRODUCTION

Every user interaction, whether conscious or unconscious, emerges as a result of complex cognitive processes (central nervous system) and motor control mechanisms (peripheral nervous system). These can be measured for example through brain activity [3, 17], heart rate variability [23], keystrokes [12], or mouse [8] and eye movements [14, 22]. To that end, in order to better understand the

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).

UIST '22 Adjunct, October 29–November 2, 2022, Bend, OR, USA

© 2022 Copyright held by the owner/author(s).

ACM ISBN 978-1-4503-9321-8/22/10.

<https://doi.org/10.1145/3526114.3558723>

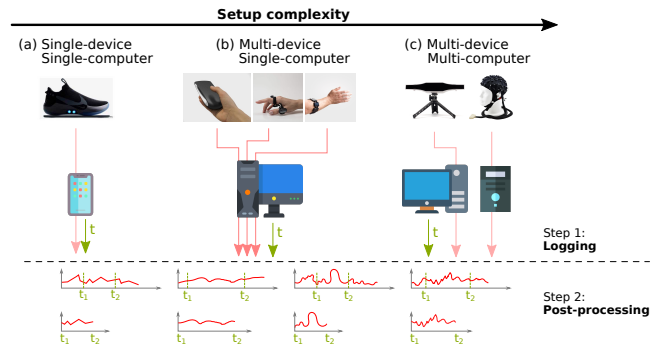


Figure 1: Gustav injects a supervisory timing signal t that helps orchestrating the experiment conditions across devices and computers, from simple (a) to complex (c) setups. Then, after post-processing, the time offsets of the recorded signals are corrected and each signal is assigned the same starting and end timestamps, thus keeping all signals in sync.

user, researchers often collect different behavioral and physiological signals simultaneously [5, 21], in order to have complementary data sources for inference and analysis. However, the collection of simultaneous data streams from multiple devices (and sometimes multiple computers) has been a longstanding challenging in HCI, neuroscience, psychology, and related areas.

If the recorded signals are not properly synchronized in time, we might draw wrong conclusions about the analyzed situation [2]. For example, while watching a video in which nothing is happening, the brain activity of the participant is reporting high values in the frequency bands associated to resting states [4] while after a few seconds those values are very low but the eye tracking data reports significant pupil dilations, which are associated with cognitive processing [6]. This will result in contradictory conclusions for the described situation, therefore temporal synchronization is critical.

We introduce Gustav,¹ a simple yet effective software-level approach for orchestrating the recording of sensory signals across devices and computers. Gustav contributes to solving three main challenges in this regard. First, it standardizes and consolidates the timeline of recorded signals based on the injection of event-related timestamps. Second, it removes unnecessary data from the recorded signals (e.g. some seconds before or after the onset of a new stimulus). Third, it adjusts the timestamp offsets of the recorded signals,

¹Named after the orchestra composer Gustav Mahler (7 July 1860 – 18 May 1911).

with millisecond precision, so that every signal coincide exactly with the duration of each experiment condition. Moreover, Gustav does not depend on any particular hardware or software program, so it can be incorporated to virtually any existing experiment design solutions such as PsychoPy,² PsychoJS,³ OpenSesame,⁴ or Paradigm.⁵ In sum, Gustav is a general and easy-to-use solution to synchronize sensory signals across devices and computers, covering thus any kind of experimental setups, as shown in Figure 1. Gustav is available as open source software.

2 RELATED WORK

Previous research has proposed to synchronize sensory signals using (1) dedicated hardware; (2) dedicated software; or (3) alignment algorithms. In this section we show that all these approaches are either vendor-locked, non-generalizable, or difficult to adopt in practice.

OpBox [7] is an integrated hardware and software toolkit for simultaneous electroencephalogram (EEG) and electromyography (EMG) signals using Matlab. The recording of other sensory signals is not supported, which makes this toolkit limited in scope. A more comprehensive solution was developed by Notaro and Diamond [9] but it is only available for the Windows Operating System.

The Observer XT [24] is also only available for Windows but it supports the design and recording of many different sensory signals. Interestingly, the user must provide visual or auditory cues to synchronize behavioral events. Automatic synchronization of physiological data can be achieved by sending out a synchronization signal through a cable that connects the computer running the stimulus presentation software and the sensory device. Other researchers have also proposed to send out a synchronization signal with dedicated hardware, such as a light-emitting diode [13] or transistor-transistor logic [20].

Ragot et al. [11] recorded cardiac and electrodermal signals and mentioned that they developed a custom program for synchronization. However, no technical details were given. Xiao et al. [19] proposed an algorithm to synchronize physiological signals, based on the direct cross-correlation of their temporal amplitudes. This method assumes that there is a common signal pattern across all the recording devices. Similarly, Wolling et al. [18] used short-time Fourier transform and Pearson's normalized cross-correlation to detect skin landmarks in order to synchronize wearable devices.

Szajerman et al. [15] synchronized EEG and eye-tracking data by detecting key events from eye movements such as blinking. Bækgaard et al. [1] proposed to do so from the EEG signal instead. Finally, Taib et al. [16] extracted key events from a driving simulator's accelerator (as it showed sharp high spikes in the recorded signal) to guide the synchronization of several heterogeneous physiological and behavioral sensors.

To the best of our knowledge, Gustav is the only offline synchronization solution that can be seamlessly integrated with any existing recording software. This makes it very convenient and

easy to use, since researchers only have to proceed with their experiments as they would normally do, and Gustav will process the acquired data once the experiment logs are stored on disk.

3 GUSTAV

Our approach for temporal synchronization of sensory signals consists of two independent components: an event logger and a file post-processor. In this section we describe each component in detail.

3.1 Event logger

The logger records event-related timestamps in Unix time format, i.e. the number of (milli)seconds elapsed since January 1, 1970. This time format is a POSIX standard. It is widely used in most Operating Systems and in all programming languages, since it provides an easy and compatible method of computing time differences. The logger is currently available in Python, JavaScript (both for the browser and for nodeJS applications), and Java.

We should mention that the events to be logged can be user-defined, although one would want to log just the beginning and end of each stimulus condition. These timestamped events are stored as a CSV file in the computer that runs the stimulus presentation software.

3.2 File post-processor

The post-processor takes as input the logged timestamped events and any number of log files from any of the recording devices. Those recorded log files (having the sensory data) should be stored as CSV files in each of the logging computers. Otherwise, we provide a conversion utility to transform tabular data in various formats (Matlab, EDF, etc.) to CSV. The post-processor performs signal synchronization in two steps: first slice the log file based on the timestamped events created by Gustav's logger, then correct the time offsets of every signal value. The post-processor is written in Python, since it is platform independent and therefore can be run in any Operating System.

Gustav can recognize automatically whether a column in a log file holds Unix timestamps. If they are not available, the post-processor can insert them automatically based on the file creation date and any timing information that is available in the log files. For example, some devices provide the number of seconds since the last calibration or since the device was turned on.

4 USE CASE EXPERIMENT

We implemented the Posner cueing task [10], a popular neuropsychological test to assess user attention. For each trial, a cue pointing right or left appears in the center of the screen, then a target will appear on the right side or left side of the screen. A participant must click on the target as fast as possible. For the experiment, we recorded EEG brain signals, mouse movements, and eye movements. The experiment setup is illustrated in Figure 2.

First, we designed the experiment in PsychoPy, which provides a graphical user interface and then generates a Python program for running the experiment. Then, we instrumented the generated Python program by inserting the Gustav logger before and after the corresponding stimulus event callbacks. Finally, after having

²<https://www.psychopy.org>

³<https://github.com/psychopy/psychojs>

⁴<https://www.opensesame.com>

⁵<http://www.paradigmexperiments.com>

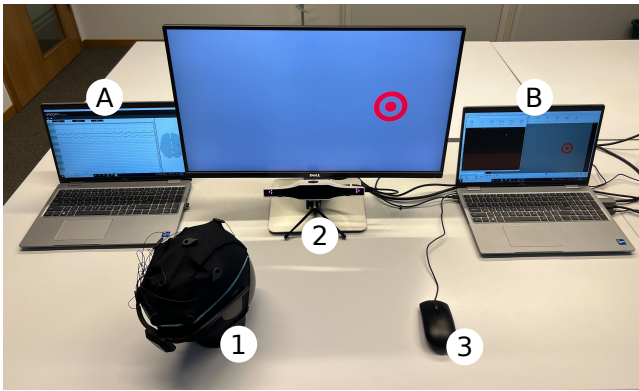


Figure 2: Arrangement of the multi-device multi-computer experiment. The brain-computer interface (1) is connected to the leftmost computer (A) which has an independent recording software (Unicorn Recorder). The eye-tracker (2) and the mouse (3) are connected to the rightmost computer (B). The recording software of the eye-tracker (Gaze-point Analysis), the mouse tracker (MouseCursorTracker Python library), and the stimulus presentation software (PsychoPy) are all running on computer B.

recorded all the signals in their respective log files, we ran the post-processor to automatically slice and correct the offsets of each log file. As a result, all recorded signals were synchronized accordingly. The supplementary video illustrates this process.

5 LIMITATIONS AND FUTURE WORK

Gustav is software-agnostic, so its event-related logging hooks must be manually added. Our goal is to automate this in the future, for which we plan to develop third party plugins for existing experiment design solutions like the ones mentioned in the Introduction: PsychoPy, OpenSesame, etc.

Gustav cannot fix hardware-related lags, though they are typically very small, often under 10 ms [16], hence below the sampling resolution we set to achieve. In addition, under the multi-computer scenario, Gustav requires that each computer has an up-to-date internal clock. Operating Systems do it by pinging public time servers over NTP,⁶ therefore as long as each computer has Internet access, this requirement is fulfilled. Note that computers do not need to be in the same timezone, they only need to be “on time.”

Finally, Gustav relies on an offline post-processing step to keep all the recorded signals in sync. While real-time synchronization is currently out of scope, it would be possible to achieve it with socket programming: all logging computers could send the data to a centralized server, be post-processed on the fly, and sent back to the logging computer before actually writing the log file.

6 CONCLUSION

Gustav is a simple yet efficient approach to orchestrating the recording of sensory signals across devices and computers, regardless the

complexity of the experimental setup, the hardware, or the stimulus presentation software. Gustav ensures that all recorded signals are in sync, with millisecond precision. It is publicly available at <https://gitlab.uni.lu/coin/gustav>.

ACKNOWLEDGMENTS

This work was supported by the Horizon 2020 FET program of the European Union through the ERA-NET Cofund funding grant CHIST-ERA-20-BCI-001 and the European Innovation Council Pathfinder program (SYMBIOTIK project).

REFERENCES

- [1] P. Bøekgaard, M. K. Petersen, and J. E. Larsen. 2014. In the twinkling of an eye: Synchronization of EEG and eye tracking based on blink signatures. In *Proc. CIP Workshop*.
- [2] A. Borawska, J. Duda, and K. Bierciewicz. 2021. Best practices of neurophysiological data collection for media message evaluation in social campaigns. *Procedia Comput. Sci.* 192 (2021).
- [3] C. de la Torre-Ortiz, M. M. Spapé, L. Kangassalo, and T. Ruotsalo. 2020. Brain relevance feedback for interactive image generation. In *Proc. UIST*.
- [4] J. C. de Munck, S. I. Gonçalves, R. Mammoliti, R. M. Heethaar, and F. H. L. da Silva. 2009. Interactions between different EEG frequency bands and their effect on alpha-fMRI correlations. *NeuroImage* 47, 1 (2009).
- [5] K. Huo, T. Wang, L. Paredes, A. M. Villanueva, Y. Cao, and K. Ramani. 2018. SynchronizAR: Instant synchronization for spontaneous and spatial collaborations in augmented reality. In *Proc. UIST*.
- [6] D. Kahneman and J. Beatty. 1966. Pupil Diameter and Load on Memory. *Science* 154, 3756 (1966).
- [7] E. Y. Kimchi, B. F. Coughlin, B. E. Shanahan, G. Piantoni, J. Pezaris, and S. S. Cash. 2020. OpBox: Open source tools for simultaneous EEG and EMG acquisition from multiple subjects. *eNeuro* 7, 5 (2020).
- [8] L. A. Leiva and R. Vivó. 2012. Interactive hypervideo visualization for browsing behavior analysis. In *Proc. WWW Companion*.
- [9] G. M. Notaro and S. G. Diamond. 2018. Simultaneous EEG, eye-tracking, behavioral, and screen-capture data during online German language learning. *Data Brief* 21 (2018).
- [10] M. I. Posner. 1980. Orienting of attention. *Q. J. Exp. Psychol. (Hove)* 32, 1 (1980).
- [11] M. Ragot, N. Martin, S. Em, N. Pallamin, and J.-M. Diverrez. 2017. Emotion Recognition Using Physiological Signals: Laboratory vs. Wearable Sensors. In *Proc. AHFE*.
- [12] M. Richardson, M. Durasoff, and R. Wang. 2020. Decoding surface touch typing from hand-tracking. In *Proc. UIST*.
- [13] E. J. Shah, J. Y. Chow, and M. J. Lee. 2020. Anxiety on Quiet Eye and Performance of Youth Pistol Shooters. *J. Sport Exerc. Psychol.* 42, 4 (2020).
- [14] Y. Sugano, X. Zhang, and A. Bulling. 2016. Aggregaze: Collective estimation of audience attention on public displays. In *Proc. UIST*.
- [15] D. Szajerman, P. Napieralski, and J.-P. Lecointe. 2018. Joint analysis of simultaneous EEG and eye tracking data for video images. In *Proc. IEEE ISEF*.
- [16] R. Taib, B. Itzstein, and K. Yu. 2014. Synchronising Physiological and Behavioural Sensors in a Driving Simulator. In *Proc. ICMI*.
- [17] C. Thanh Vi, K. Hornbæk, and S. Subramanian. 2017. Neuroanatomical correlates of perceived usability. In *Proc. UIST*.
- [18] F. Wollong, C. D. Huynh, and K. Van Laerhoven. 2021. IBSync: Intra-body synchronization of wearable devices using artificial ECG landmarks. In *Proc. ISWC*.
- [19] R. Xiao, C. Ding, and X. Hu. 2022. Time Synchronization of Multimodal Physiological Signals through Alignment of Common Signal Types and Its Technical Considerations in Digital Health. *J. Imaging* 8, 5 (2022).
- [20] J. Xue, C. Quan, C. Li, J. Yue, and C. Zhang. 2017. A crucial temporal accuracy test of combining EEG and Tobii eye tracker. *Medicine* 96, 13 (2017).
- [21] D. Yoon, N. Chen, F. Guimbretière, and A. Sellen. 2014. RichReview: blending ink, speech, and gesture to support collaborative document review. In *Proc. UIST*.
- [22] X. Zhang. 2021. Evaluating the Effects of Saccade Types and Directions on Eye Pointing Tasks. In *Proc. UIST*.
- [23] Y. Zhao, B. Li, Y. Li, J. Zhou, J. Cao, Y. Luo, X. Zhou, C. Yao, L. Shi, and G. Wang. 2021. Rope X: Assistance and Guidance on Jumping Rope Frequency, based on Real-time, Heart Rate Feedback During Exercise. In *Adj. Proc. UIST*.
- [24] P. H. Zimmerman, J. E. Bolhuis, A. Willemsen, E. S. Meyer, and L. P. Noldus. 2009. The Observer XT: A tool for the integration and synchronization of multimodal signals. *Behav. Res. Methods* 41, 3 (2009).

⁶NTP is a networking protocol for clock synchronization over packet-switched variable-latency data networks.