The Journal of Open Source Software

Octopus Sensing: A python library for human behavior studies

Nastaran Saffaryazdi[1], **Aidin Gharibnavaz**[3], **and Mark Billinghurst**[12]

1 Empathic Computing Laboratory, Auckland Bioengineering Institute, University of Auckland **2** Empathic Computing Laboratory, University of South Australia **3** Independent Researcher

DOI: 10.21105/joss.04045

Software

- Review
- Repository
- Archive

Editor: Andrew Stewart

Reviewers:

- @ixjlyons
- @peircej

Submitted: 15 December 2021
Published: 14 March 2022

Summary

Designing user studies and collecting data is critical to exploring and automatically recognizing human behavior. It is currently possible to use a range of sensors to capture heart rate, brain activity, skin conductance, and a variety of different physiological cues (Seneviratne et al., 2017). These data can be combined to provide information about a user's emotional state (Dzedzickis et al., 2020; Egger et al., 2019), cognitive load (Mangaroska et al., 2022; Vanneste et al., 2021), or other factors. However, even when data are collected correctly, synchronizing data from multiple sensors is time-consuming and prone to errors. Failure to record and synchronize data is likely to result in errors in analysis and results, as well as the need to repeat the time-consuming experiments several times. To overcome these challenges, `Octopus Sensing` facilitates synchronous data acquisition from various sources and provides some utilities for designing user studies, real-time monitoring, and offline data visualization. The primary aim of `Octopus Sensing` is to provide a simple scripting interface so that people with basic or no software development skills can define sensor-based experiment scenarios with less effort.

Statement of need

Several changes occur in the body and mind due to various internal and external stimuli. Nowadays, researchers use various sensors to measure and monitor these responses to determine an individual's state (Chen et al., 2021; Kreibig, 2010; Sun et al., 2020) and to assist patients (Hassouneh et al., 2020) or monitor mental health (Jiang et al., 2020). Monitoring and analyzing human responses can be used to improve social interactions (Hossain & Gedeon, 2019; Verschuere et al., 2006) and improve quality of life by creating intelligent devices such as Intelligent Tutoring Systems (Dewan et al., 2019), creating adaptive systems (Aranha et al., 2019), or creating interactive robots and virtual characters (Hong et al., 2021; Val-Calvo et al., 2020).

Researchers have recently attempted to gain a deeper understanding of humans by simultaneously studying physiological and behavioral changes in the human body (Koelstra et al., 2011; Shu et al., 2018). Acquiring and analyzing data from different sources with various hardware and software is complex, time-consuming, and challenging. Additionally, human error can easily affect synchronously recording data in multiple formats. These tasks slow down the pace of progress in human-computer interaction and human behavior research.

There are only a few frameworks that support synchronous data acquisition and design. iMotions has developed software for integrating and synchronizing data recording through a wide range of various sensors and devices. Despite having many great features, iMotions is commercial software and not open-source. In contrast, there are a few open-source programs for conducting human studies. Psychopy (Peirce et al., 2019) is a powerful open-source, cross-platform software that is mainly used for building experiments' pipelines in behavioral science with visual and auditory stimuli. It can also record data from a few devices and send triggers to them. Another effort in this area is LabStreamingLayer (LSL) LabRecorder. Although LSL LabRecorder provides synchronized, multimodal streaming through a wide range of devices, an extra application still needs to be run for acquiring data from each sensor separately.

`Octopus Sensing` is a lightweight open-source multi-platform library that facilitates synchronous data acquisition from various sources through a unified interface and could be easily extended to process and analyze data in real-time. We designed the `Octopus Sensing` library to minimize the effect of network failure in synchronous data streaming and reduce the number of applications that we should run for data streaming through different devices. Rather than creating a standalone software or framework, we created a library that could be easily integrated with other applications. `Octopus Sensing` provides a real-time monitoring system for illustrating and monitoring signals remotely using a web-based platform. The system also offers offline data visualization to see various human responses simultaneously.

Overview

`Octopus Sensing` is a tool to help in running scientific experiments that involve recording data synchronously from multiple sources. It can simultaneously collect data from various devices such as OpenBCI EEG headset, Shimmer3 sensor, camera, and audio-recorder without running another software for data recording. Data recording can be started, stopped, and triggered synchronously across all devices through a unified interface.

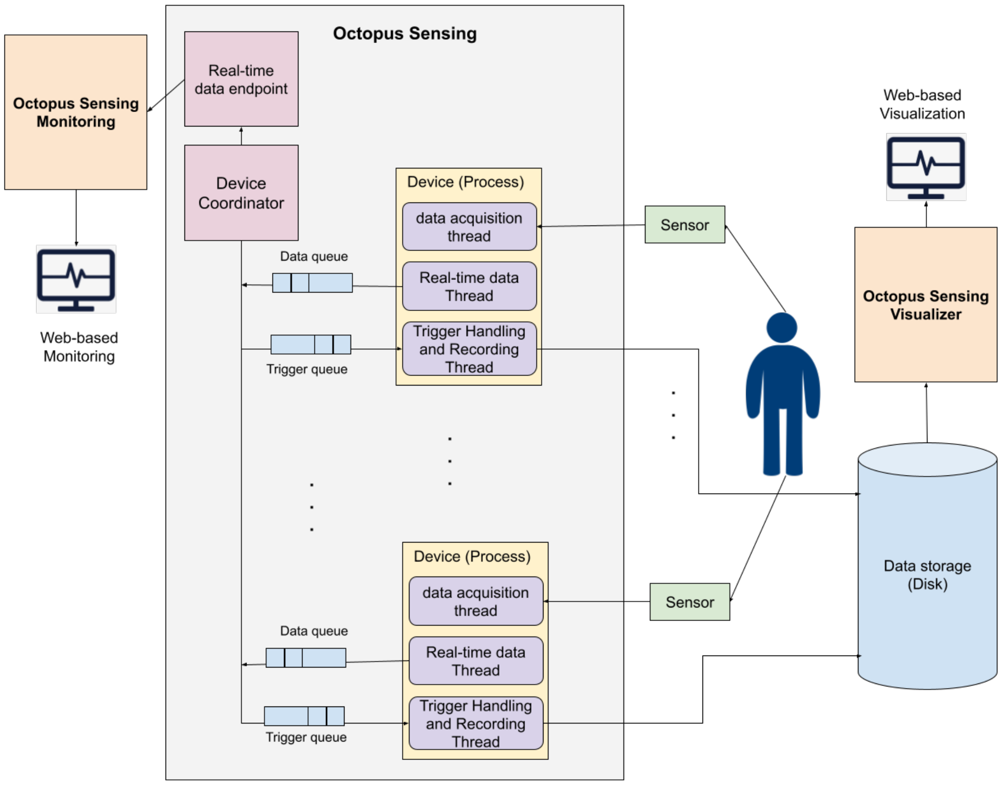
The main features of `Octopus Sensing` are listed as follows:

- Manages data recording from multiple sources using a simple unified interface.
- Minimizes human errors from manipulating data in synchronous data collection.
- Provides some utilities for designing studies like showing different stimuli or designing questionnaires.
- Offers a monitoring interface that prepares and visualizes collected data in real-time.
- Provides offline visualization of data from multiple sources simultaneously.

Methodology

`Octopus Sensing` synchronizes data recording by using `multiprocessing` in Python. By instantiating the `Device` class, `Octopus Sensing` creates a process for the device. Each device's process has three threads: `data acquisition` thread, `trigger handling and data recording` thread, and `real-time data` thread. The `data acquisition` thread is responsible for acquiring data from a sensor. The `trigger handling and data recording` thread handles trigger messages through a message queue for synchronous data recording. It also records data in a file or files. The `real-time data` thread listens on a queue for requests and returns the last three seconds of the recorded data in the same queue. This data is being used in real-time monitoring and can be used for real-time processing and creating real-time feedback in the future. The `Device Coordinator` sends different triggers such as the start of recording or end of recording to different devices by putting the message in all devices' trigger queues at the same time. The `Device Coordinator` can also send the trigger over the network for devices that are not embedded in the `Octopus Sensing`. The following image shows the overall view of the main components of the `Octopus Sensing` and their relations.

The Journal of Open Source Software



-

Research perspective

We used `Octopus Sensing` to design several human emotion recognition user studies. We developed a user study using `Octopus Sensing` for recording facial video, brain activity, and physiological signals during a watching video task. The recorded data was used to make multimodal emotion recognition models (Saffaryazdi, Wasim, et al., 2022). This scenario which is common in physiological emotion recognition studies has been included in the repository as an example and explained in the tutorial. In another study, we collected multimodal data during face-to-face conversations to make models for emotion recognition during interactive tasks (Saffaryazdi, Goonesekera, et al., 2022). We developed this user study's scenario in Python using the `Octopus Sensing` library and conducted the study only by running our developed program, without running any other software or any supervision for data recording or data synchronization.

This tool can be used to build real-time data processing systems to recognize emotions, stress, cognitive load, or analyze human behavior. Our final goal is to extend its capabilities to provide real-time emotion recognition using multimodal data. Furthermore, we plan to integrate it with `Psychopy` in the future and combine multimodal data collection and monitoring with `Psychopy` features when designing scenarios. Additionally, we plan to support LSL in the future. By supporting LSL, other applications that already support LSL could work with `Octopus Sensing`.

Acknowledgement

We acknowledge the Empatic Computing Laboratory (ECL) for financial support and for providing feedback, and Professor Suranga Nanayakkara for encouragement and feedback.

References

Aranha, R. V., Corrêa, C. G., & Nunes, F. L. (2019). Adapting software with affective computing: A systematic review. *IEEE Transactions on Affective Computing*, *12*(4), 883–899. https://doi.org/10.1109/TAFFC.2019.2902379

Chen, K.-H., Brown, C. L., Wells, J. L., Rothwell, E. S., Otero, M. C., Levenson, R. W., & Fredrickson, B. L. (2021). Physiological linkage during shared positive and shared negative emotion. *Journal of Personality and Social Psychology*, *121*(5), 1029. https://doi.org/10.1037/pspi0000337

Dewan, M. A. A., Murshed, M., & Lin, F. (2019). Engagement detection in online learning: A review. *Smart Learning Environments*, *6*(1), 1. https://doi.org/10.1186/s40561-018-0080-z

Dzedzickis, A., Kaklauskas, A., & Bucinskas, V. (2020). Human emotion recognition: Review of sensors and methods. *Sensors*, *20*(3), 592. https://doi.org/10.3390/s20030592

Egger, M., Ley, M., & Hanke, S. (2019). Emotion recognition from physiological signal analysis: A review. *Electronic Notes in Theoretical Computer Science*, *343*, 35–55. https://doi.org/10.1016/j.entcs.2019.04.009

Hassouneh, A., Mutawa, A., & Murugappan, M. (2020). Development of a real-time emotion recognition system using facial expressions and EEG based on machine learning and deep neural network methods. *Informatics in Medicine Unlocked*, *20*, 100372. https://doi.org/10.1016/j.imu.2020.100372

Hong, A., Lunscher, N., Hu, T., Tsuboi, Y., Zhang, X., Franco dos Reis Alves, S., Nejat, G., & Benhabib, B. (2021). A multimodal emotional human–robot interaction architecture for social robots engaged in bidirectional communication. *IEEE Transactions on Cybernetics*, *51*(12), 5954–5968. https://doi.org/10.1109/TCYB.2020.2974688

Hossain, M. Z., & Gedeon, T. (2019). Observers' physiological measures in response to videos can be used to detect genuine smiles. *International Journal of Human-Computer Studies*, *122*, 232–241. https://doi.org/10.1016/j.ijhcs.2018.10.003

Jiang, Y., Li, W., Hossain, M. S., Chen, M., Alelaiwi, A., & Al-Hammadi, M. (2020). A snapshot research and implementation of multimodal information fusion for data-driven emotion recognition. *Information Fusion*, *53*, 209–221. https://doi.org/10.1016/j.inffus.2019.06.019

Koelstra, S., Muhl, C., Soleymani, M., Lee, J.-S., Yazdani, A., Ebrahimi, T., Pun, T., Nijholt, A., & Patras, I. (2011). Deap: A database for emotion analysis; using physiological signals. *IEEE Transactions on Affective Computing*, *3*(1), 18–31. https://doi.org/10.1109/T-AFFC.2011.15

Kreibig, S. D. (2010). Autonomic nervous system activity in emotion: A review. *Biological Psychology*, *84*(3), 394–421. https://doi.org/10.1016/j.biopsycho.2010.03.010

Mangaroska, K., Sharma, K., Gašević, D., & Giannakos, M. (2022). Exploring students' cognitive and affective states during problem solving through multimodal data: Lessons learned from a programming activity. *Journal of Computer Assisted Learning*, *38*(1), 40–59. https://doi.org/10.1111/jcal.12590

Peirce, J., Gray, J. R., Simpson, S., MacAskill, M., Höchenberger, R., Sogo, H., Kastman, E., & Lindeløv, J. K. (2019). PsychoPy2: Experiments in behavior made easy. *Behavior Research Methods*, *51*(1), 195–203. https://doi.org/10.3758/s13428-018-01193-y

Saffaryazdi, N., Goonesekera, Y., Saffaryazdi, N., Hailemariam, N. D., Temesgen, E. G., Nanayakkara, S., Broadbent, E., & Billinghurst, M. (2022). *Emotion recognition in conversations using brain and physiological signals*. In press. https://doi.org/10.1145/3490099.3511148

Saffaryazdi, N., Wasim, S. T., Dileep, K., Farrokhinia, A., Nanayakkara, S., Broadbent, E., & Billinghurst, M. (2022). *Using facial micro-expressions in combination with EEG and physiological signals for emotion recognition*. Under review.

Seneviratne, S., Hu, Y., Nguyen, T., Lan, G., Khalifa, S., Thilakarathna, K., Hassan, M., & Seneviratne, A. (2017). A survey of wearable devices and challenges. *IEEE Communications Surveys & Tutorials*, *19*(4), 2573–2620. https://doi.org/10.1109/COMST.2017.2731979

Shu, L., Xie, J., Yang, M., Li, Z., Li, Z., Liao, D., Xu, X., & Yang, X. (2018). A review of emotion recognition using physiological signals. *Sensors*, *18*(7), 2074. https://doi.org/10.3390/s18072074

Sun, Y., Ayaz, H., & Akansu, A. N. (2020). Multimodal affective state assessment using fNIRS+ EEG and spontaneous facial expression. *Brain Sciences*, *10*(2), 85. https://doi.org/10.3390/brainsci10020085

Val-Calvo, M., Álvarez-Sánchez, J. R., Ferrández-Vicente, J. M., & Fernández, E. (2020). Affective robot story-telling human-robot interaction: Exploratory real-time emotion estimation analysis using facial expressions and physiological signals. *IEEE Access*, *8*, 134051–134066. https://doi.org/10.1109/ACCESS.2020.3007109

Vanneste, P., Raes, A., Morton, J., Bombeke, K., Van Acker, B. B., Larmuseau, C., Depaepe, F., & Van den Noortgate, W. (2021). Towards measuring cognitive load through multimodal physiological data. *Cognition, Technology & Work*, *23*(3), 567–585. https://doi.org/10.1007/s10111-020-00641-0

Verschuere, B., Crombez, G., Koster, E., & Uzieblo, K. (2006). Psychopathy and physiological detection of concealed information: A review. *Psychologica Belgica*, *46*(1-2), 99. https://doi.org/10.5334/pb-46-1-2-99