The Journal of Open Source Software

OpenHPS: A Modular Framework to Facilitate the Development and Sharing of Positioning Systems and Algorithms

Maxim Van de Wynckel [1] **and Beat Signer** [1]

1 Vrije Universiteit Brussel, Belgium

DOI: 10.xxxxxx/draft

Software

- Review
- Repository
- Archive

Editor: Open Journals

Reviewers:

- @openjournals

Submitted: 01 January 1970
Published: unpublished

Summary

Positioning systems are a collection of technologies and algorithms that can help to determine the location of people and objects. While outdoor positioning systems mainly use satellites such as GPS to perform the tracking, many different technologies exist that can offer better latency, accuracy or battery efficiency depending on the use case. These use cases can range from indoor positioning systems where there is no standardised technology to smaller scale positioning systems such as tracking the location of a pen on a piece of paper. In the research domain of positioning and tracking, these novel types of positioning systems are often implemented as single use protypes with no common data format, making it difficult to expand. OpenHPS was created as a framework to enable to creation of positioning systems for a wide range of use cases and to also allow a wide range of algorithms. The creation of a positioning system using our framework is graph based, allowing developers to share *nodes* that implement new algorithms or combine various technologies.

Statement of Need

Context

OpenHPS is an open source hybrid positioning system framework written in TypeScript. It can be run on the server using NodeJS, the browser or even hybrid mobile applications such as NativeScript, React-Native and CapacitorJS. The general design of a positioning system created using our framework consists of a graph with a set of nodes that process data.

All concepts, ranging from positions to sensor values can be expressed in various ways with varying units, allowing OpenHPS to be used for small scale use cases such as tracking a pen on a paper to larger use cases such as tracking airplanes across the globe.

Related Work

Framework

Data Frames and Objects

Our framework uses stream based processing of `DataFrames` which contain all time-sensitive information. Inside these frames can be one or more `DataObjects` which indicate the spatial objects that are relevant for the information within the data frame.

Both this information can be serialised to JSON or semantic RDF data, enabling the interoperability between systems.

Nodes

Each node in the graph represent a step in the processing of data frames from source to sink. A `SourceNode` generates information, such as a sensor that .

Communication

Modules

OpenHPS is modular by design, mainly due to the ability to extend data frames and objects. These extensions allow for different data objects such as different sensors such as cameras and IMU sensors or spatial landmarks such as Bluetooth beacons.

Each node in a positioning model can be extended as well, allowing the creation of custom algorithms that can be added or removed from a positioning system with ease.

Performance

Since our framework uses TypeScript, it uses JavaScript at runtime, which is single-threaded by default. To overcome the challenges associated with this when creating real time stream processing systems, our graphs or portions of the graph can be run with the help of web workers. All data transmitted through our graph is serialisable, which prevents the need for developers to handle this serialisation or the communication between web workers themselves.

In addition to the ability to run graphs on multiple workers, our communication nodes such as MQTT, combined with the serialisability of data, allows developers to offload the processing of complicated tasks to other servers or dedicated processors. For more high-demanding algorithms such as computer vision and visual SLAM, modules such as **(openhps/opencv?)** and **(openhps/openvslam?)** create C++ bindings in NodeJS.

Examples of Rearch Work

OpenHPS has been a building block for various research such as its use within indoor positioning systems (Van de Wynckel & Signer, 2021), its ability to serialise location data to RDF in a demonstrator application that aims to preserve privacy and transparency using Solid-pods (Van de Wynckel & Signer, 2022) and its use within the SemBeacon demonstrator application (Van de Wynckel & Signer, 2023) that is written in CapacitorJS and uses OpenHPS to deserialise positioning data and positioning systems. However, with the ability for OpenHPS to contain modular nodes, its use within the domain of positioning systems can allow for easier sharing of algorithms and findings, as well as the rapid creation of prototypes and demonstrators that make use of location data.

References

Van de Wynckel, M., & Signer, B. (2022). A Solid-based Architecture for Decentralised Interoperable Location Data. *Proceedings of IPIN 2022, The 12th International Conference on Indoor Positioning and Indoor Navigation, CEUR Workshop Proceedings*, 1–15.

Van de Wynckel, M., & Signer, B. (2021). Indoor Positioning Using the OpenHPS Framework. *Proceedings of IPIN 2021, The 11th International Conference on Indoor Positioning and Indoor Navigation*, 1–8. https://doi.org/10.1109/IPIN51156.2021.9662569

Van de Wynckel, M., & Signer, B. (2023). SemBeacon: A Semantic Proximity Beacon Solution for Discovering and Detecting the Position of Physical Things. *Proceedings of IoT 2023, The 13th International Conference on the Internet of Things*, 1–8.