November 2022

Utilizing Haptics Vibrations to Determine Spatial Context

D Shin

Utilizing Haptics Vibrations to Determine Spatial Context

Abstract:

This publication describes techniques and systems for utilizing haptics vibrations to determine a spatial context for a computing device. In an aspect, a system of determining a spatial context utilizes haptics vibrations that are measured by an on-board inertial measurement unit (IMU). The measured haptics sensor data are input into a machine-learned (ML) model (e.g., an embedding model). The embedding model utilizes gravity removal, matched filtering, envelope filtering, and convolutional embedding to generate a spatial context prediction.

Keywords:

Haptics vibration, haptics motor, haptics, inertial measurement unit (IMU), feature compressor, envelope filtering, matched filtering, spatial context, buffer, embedding, linear resonant actuator (LRA), convolutional embedding, machine-learning, ML model

Background:

A smartphone can determine a context within its own physical surroundings (e.g., a spatial context) and utilize the determined spatial context to implement its functions and features. For example, based on spatial context, a smartphone device can recognize if the user is holding the phone in either a landscape mode or a portrait mode, enabling the device to change a display orientation (e.g., display angle). A conventional method of determining the spatial context on smartphones is through detecting the 6-axis orientation of an inertial measurement unit (IMU).

However, such measurements are of limited use for determining spatial context, often making it difficult to detect whether a smartphone is being held horizontally by the user or sitting horizontally on a desk, for example. As a result, there is a need for improved methods of determining spatial contexts around smartphones.

Description:

This publication describes techniques and systems (collectively, just "techniques") for utilizing haptics vibrations (e.g., haptics sensor data) to determine a spatial context for a computing device (e.g., a smartphone). An example computing device includes a processor, a haptics motor, and a haptics vibration sensor. The haptics motor (e.g., linear resonant actuator (LRA)) provides haptic feedback to a user. A triggering event, like a phone call being received, may trigger the haptics motor to generate a vibration that indicates to the user that they are getting a phone call. The haptics vibration sensor (e.g., an inertial measurement unit (IMU), a microphone) senses vibrations generated by a haptics motor and generates haptics sensor data. In aspects, the haptics vibration sensor is a 6-axis IMU (3-axis gyroscope and 3-axis accelerometer).

The computing device also includes a computer-readable medium (CRM). The CRM may include any suitable memory or storage device. Device data (e.g., haptics sensor data, a computer program module) is stored on the CRM. The device data includes program instructions for one or more computer program modules (e.g., applications, an operating system) executable by the processor to provide the functionality described herein. The term "module" refers to computer program logic (e.g., program instructions) used to provide the specified functionality. Thus, a module can be implemented in hardware, firmware, and/or software. The computer program modules in the device data include a Spatial Context module.

The CRM also includes a machine-learned model (ML model). The ML model may be a standard neural-network-based model with corresponding layers required for processing input features like fixed-side vectors, text embeddings, or variable length sequences. The ML model may be implemented as one or more of a support vector machine (SVM), a recurrent neural network (RNN), a convolutional neural network (CNN), a dense neural network (DNN), one or more heuristics, other machine-learning techniques, a combination thereof, and so forth.

In aspects, the ML model is an embedding model that receives, as inputs, haptics sensor data. The embedding model is trained to classify data clusters to determine where a smartphone is located and generate predictions for the spatial context of the smartphone. The haptics sensor data collected is unique to different spatial contexts (e.g., phone in a bag, phone in hand).

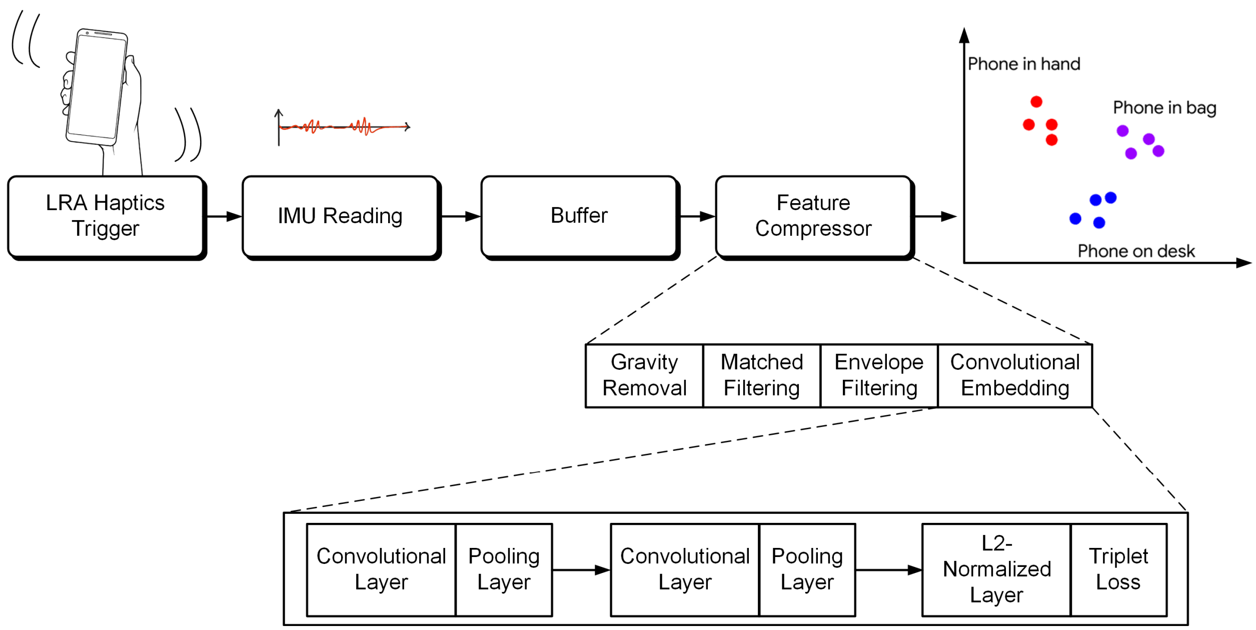

Figure 1

Figure 1 is a schematic illustration of an algorithmic flow of an example technique for utilizing haptics vibrations to determine a spatial context for a computing device (e.g., smartphone). Aspects of the algorithmic flow may be implemented by the embedding module and/or the Spatial Context module of the smartphone. The figure illustrates a smartphone held in the hand of a user. As a result of a triggering event (e.g., a phone call received), the haptics motor (e.g., the LRA) of the smartphone is driven with a custom burst train waveform that repeats a short pulse of haptics N times (e.g., two times for a notification). The haptics vibration sensor (e.g., an IMU) records raw haptics sensor data. The raw haptics sensor data can be stored in the CRM in its raw and unaltered format, for example, in a buffer.

The buffered haptics sensor data can then be provided to a machine-learned model (e.g., a feature compressor) as an input. The feature compressor has four separate components that are serially run on the buffered haptics sensor data to output a final scalar feature corresponding with band fit. The first component of the feature compressor is a gravity removal algorithm that removes all static (e.g., direct current (DC)) components present in the buffered haptics sensor data, including the removal of haptics sensor data that has no information about the transient motion caused by the haptics motor. Gravity removal can be performed utilizing a low-order high-pass filter on the haptics sensor data to remove all frequencies that are close to zero.

The second component of the feature compressor is a matched filter that performs pulse train matched filtering. In pulse train matched filtering, each channel of the haptics sensor data is correlated with the original haptics waveform to emulate a “matched filter” effect, enabling the detection of the presence of an unknown signal. The time in the correlation when the original haptics waveform and the unknown signal perfectly align will have the highest amplitude. The detected unknown signal can then be removed.

The third component of the feature compressor is an envelope filter that performs envelope filtering. Envelope filtering is done to even out the result of the pulse train matched filtering, which is likely to have multiple spikes of different frequencies. The envelope filter will only filter through the low frequency signals that have no high-resolution interference. With the envelope filter, the amplitude features (matched filtered envelopes) are the only signals that are used.

The fourth and final component of the feature compressor is convolutional embedding performed by a convolutional embedding network (the "embedding model"). The matched filtered envelopes are passed through the embedding model. Once the matched filtered envelopes pass through the two standard convolutional and pooling layers, they are processed through a L2-normalized layer and a triplet loss layer, where they are transformed into individual scalar data points that can be plotted on a graph corresponding with band fit. Figure 1 illustrates this as a graph of multiple data points in data clusters that predict the spatial context (e.g., location) of the smartphone as one of located in a user's hand, located on a desk, or located in a bag.

Responsive to a spatial context prediction generated by the embedding model, the Spatial Context module may change a configuration or setting of the smartphone. For example, upon determining that the smartphone is located in a bag, the Spatial Context module may enable a power-saving mode for the device. In another example, after finding that the smartphone is located on a desk, the Spatial Context module may initiate a background data refresh.

The embedding model may be trained during a training phase based on haptics sensor data that is collected and categorized with corresponding correct haptics classifications (e.g., hand touching device, device in a bag, device on a table). The training can be performed when the device is offline using multiple lab datasets so that during the deployment of the embedding model, the smartphone can check which of the previously identified clusters the runtime embedding is closest to. After sufficient training, the embedding model can be deployed to the CRM of a computing device.

In a first example use case, assume that a user sets their smartphone down in their work bag (e.g., a purse). Using the haptics motor to vibrate and sense its surroundings, the Spatial Context module can determine that the spatial context for the smartphone is a location inside a bag. Utilizing the spatial context, the Spatial Context module can cause the smartphone to enter into a power-saving mode. In a second example use case, assume that a user places their smartphone on a flat planar surface (e.g., a desk, a table). After the Spatial Context module is deployed, the smartphone determines that it is on a flat surface, which, in turn, causes the smartphone to perform a background data refresh (e.g., checking email, updating applications).

References:

[1] Ali, Kamran and Liu, Alex X. "Fine-grained Vibration Based Sensing Using a Smartphone." August 27, 2020. https://arxiv.org/pdf/2007.03874.pdf.

[2] Cho, Jungchan, Hwang, Inhwan, and Oh, Songhwai. "Vibration-Based Surface Recognition for Smartphones." Accessed (August 19, 2022). http://rllab.snu.ac.kr/publications/papers/2012_cpsna_vibephone.pdf. Published at least as early as January 19, 2021. Internet Archive. https://web.archive.org/web/20210115000000*/http://rllab.snu.ac.kr/publications/papers/2012_cpsna_vibephone.pdf.

[3] Patent Publication: US20180348867A1. Method to Sense Force Exertion on Handheld Device for Behavior Setting and Input for Quick Response. Priority Date: June 5, 2017.

[4] Patent Publication: US20190087007A1. Providing Haptic Output Based on a Determined Orientation of an Electronic Device. Priority Date: May 21, 2014.

[5] Kerzel, Matthias, Ali, Moaaz, Ng, Hwei Geok, and Wermter, Stefan. "Haptic Material Classification with a Multi-Channel Neural Network." Institute of Electrical and Electronics Engineers. May 2017. https://www2.informatik.uni-hamburg.de/wtm/publications/2017/KANW17/Kerzel-Haptic_Material_Classification_2017-Webpage.pdf.