Virtual sensors - Synthesizing dynamic crowdsensing data into information on static instances

Ligtvoet, Birgit R.; Verbree, Edward; Gorte, Ben G.H.

DOI
10.3929/ethz-b-000225584

Publication date
2018

Document Version
Final published version

Published in
Adjunct Proceedings of the 14th International Conference on Location Based Services

Virtual sensors - Synthesizing dynamic crowdsensing data into information on static instances

Author(s):
Ligtvoet, Birgit R.; Verbree, Edward; Gorte, Ben G.H.

Publication Date:
2018-01-15

Permanent Link:
https://doi.org/10.3929/ethz-b-000225584 →

 http://doi.org/10.3929/ethz-b-000225584

Virtual sensors - Synthesizing dynamic crowdsensing data into information on static instances

B.R. Ligtvoet*, E. Verbree**, and B.G.H. Gorte***

* Delft University of Technology, MSc Geomatics for the Built Environment, Julianalaan 138, Delft, The Netherlands, bligtvoet@icloud.com
** Delft University of Technology, Department OTB - Research for the Built Environment, Julianalaan 138, Delft, The Netherlands, e.verbree@tudelft.nl
*** Delft University of Technology, Department of Geoscience and Remote Sensing, Stevinweg 1, Delft, The Netherlands, b.g.h.gorte@tudelft.nl

Abstract. There has been a steady increase in applications that rely on crowdsensing to gather data for analysis purposes. Crowdsensing enables the use of dynamic sensors to collect data on static objects of interest. However, using dynamic sensors in this way causes a problem. The focus of the collected data is on the position of the sensor, not on the object of interest. This results in difficulties in tracking the object of interest in terms of what part of the data from the dynamic sensor describes the object of interest. To shift the focus from the dynamic sensors to a static object, the virtual sensor is introduced. A virtual sensor enables the grouping of data from different dynamic sensors into a single virtual sensor based on measurement positions. The data from the multiple dynamic sensors can be analyzed to provide information per virtual sensor. The data structure of a visual sensor is close to the SensorThings API data structure, which can be expanded to support virtual sensors by adding an additional entity.

1. Introduction

With the surge in smartphone popularity, there has been an increase in applications that rely on crowdsensing to gather data. Crowdsensing allows researchers to make use of a pre-existing network of hardware with a large coverage to gather large amounts of data on static objects. This type of crowdsensing can be used as an alternative to tracking through Wi-Fi with the advantage of not having to own and maintain hardware or to monitor objects that do not allow the direct placement of sensors. One example, which will be further addressed in a case study, is the use of smartphones to gather data on road pavement distresses. The type of crowdsensing used is the opportunistic kind, the owner of the smartphone does not need to perform any actions after he or she has installed the application used to collect data (Ganti et al, 2011).

When using crowdsensing to gather data on road pavement distresses, the sensor is dynamic while the object of interest is static. The usage of dynamic sensors in combination with static instances involves two challenges. First, crowdsensing relies on verification through consistency. The data gathered through crowdsensing gains value when measurements from multiple sources show similar results. Second is the structuring of the data through standards in combination with the dynamic sensors. Sensor standards automatically put the focus on the position of the sensor instead of the position of the object of interest. The focus on the sensor in these standards makes the tracking of the sensor position a simple task. However, combining the information describing the same object of interest from multiple sensors requires additional work.

The above factors show that the current state of structuring data and applying sensor standards is not ideal when using crowdsensing in combination with dynamic sensors and static objects of interest.

2. The virtual sensor

The term virtual sensor is often used to describe non-physical sensors that:

- use data from real sensors to replace temporary sensors;
- increase the robustness of sensor systems in case of sensor failure;
- provide a continuous output from non-continuous sensors;
- and predict sensor values (Wilson, 1997)

A virtual sensor needs to perform at least one of the above tasks. In the case of crowdsensing and dynamic sensors measuring the state of static objects, the goal is to gather data from multiple sensors of the same type into one virtual sensor. The real sensors used to gather the data calculate their position separately due to having separate Global Navigation Satellite System receivers. The virtual sensor is used to group together data with a similar position within a pre-set timeframe.

In short, the attributes of a virtual sensor used for crowdsensing are as follows:

 http://doi.org/10.3929/ethz-b-000225584

- a virtual sensor only exists when there are data points positioned near each other;
- a virtual sensor groups together data points from different real sensors based on their position relative to each other;
- a virtual sensor has a static position while the real sensor used to collect data is dynamic;
- the validity of a virtual sensor is dependent on the time interval at which it is created;
- the position of a virtual sensor is the average of the positions of the data grouped by the virtual sensor;
- a virtual sensor can be used to calculate averages based on the grouped data;
- and a virtual sensor does not support historical consistency between time-frames.

A virtual sensor can group together multiple types of data i.e. single detections but also short time series. In the case that the data to be grouped can be seen as a group in itself, the center of the time series is used as the position which is used in the position calculation for the virtual sensor itself.

3. Virtual sensors and sensor standards

Structuring virtual sensors to fit standards provides a clear documentation for other parties interested in the data collected by the virtual sensor, providing the opportunity to use data from virtual sensors and crowdsensing for other purposes as well as simplifying the way people can interact with the data. In this paper, the alignment to the OGC SensorThings API is studied because of its focus on open source, interoperability and ability to connect to other standardized GIS-based services (Liang et al, 2016).

There are many similarities between the way a virtual sensor can be structured and the SensorThings API. However, the virtual sensor cannot completely conform to the sensor standard as it is currently defined. The misalignment is located in the DataStream and MultiDataStream entities and their relationship to the Observation entity. In the case of using the DataStream entity, the Observation can only be one measurement, not a group of measurements. Grouping is done in the DataStream entity and DataStream cannot exist of multiple other DataStreams.

The MultiDataStream allows the Observation entity to send an array of data consisting of multiple different types of measurements gathered at the same time instance. The Observation can contain information on different phe-

http://doi.org/10.3929/ethz-b-000225584

nomena like the position, temperature, and altitude but not a time series of only one phenomenon.

Virtual sensors created through using crowdsensing can be made compatible by creating an additional instance of a DataStream called a VsDatastream, while the original Datastream entity is renamed to RsDatastream. The newly created VsDatastream allows for the second grouping phase needed to move from a dynamic real sensor to a static virtual sensor. The VsDatastream and RsDatastream entities visualize the concept of verification through consistency. The VsDatastream of the virtual sensor needs to contain multiple instances of the RsDatastream entity from a real sensor.

4. Case study - Virtual sensors for Road Asset Management

The case study on how to create a virtual sensor is focussed on road Asset Management and specifically on monitoring the state of the pavement of a road section. Modern contracts in the building and infrastructure sector are forcing contractors to focus on efficient maintenance. However, the contractors face the problem of having no up-to-date information on the state of the road due to the limited availability of specialized hardware (Klunder et al, 2010). This leaves contractors in a state of uncertainty on where road pavement distresses are occurring and how severe these distresses are. Crowdsensing through smartphones inside car cabins can be used to collect a large amount of low-quality data on the state of the road pavement. During the case study data was gathered by using up to 4 smartphones (iPhone 5, iPhone 5c, iPhone 5s and iPhone 6) simultaneously in both a public transport bus type Volvo 8900 and a BMW 523i.

4.1. The relationship between sensor and road pavement distress

To use crowdsensing in a successful manner, the effects of road pavement distresses on smartphone sensor outputs in a moving vehicle have to be studied. During the case study, the vertical acceleration in combination with the position of the smartphone were the main sources of information. However, the phenomenon being measured is the vertical acceleration the car is exerting on the smartphone inside the car cabin caused by road pavement distresses. This relationship puts a major emphasis on the position of a dynamic object, the car, instead of the object of interest, the road pavement distress. The ideal situation consists of a sensor that is equipped directly to the road pavement distress located somewhere along the road.

To shift the focus from cars to road pavement distress, the notion that measuring something through crowdsensing at a location is the same as measuring through a sensor embedded in the pavement at the same location is used. The sensor embedded in the road pavement would be a virtual sensor; it does not exist.

4.2. Creating the virtual sensor

The next step is to design how the raw data from the smartphone can be ordered in a way that it can be connected to road pavement distress through virtual sensors. The data used to create a virtual sensor consists of time series called events. An event consists of arrays of vertical acceleration measurements with timestamps and positions. An event is created when measurement values exceed a threshold value that indicates a road pavement distress as the cause of the measurement. For example, if a measurement is outside 2.75 standard deviation, then it is most likely caused by a road pavement distress. The created event is comprised of both the measurements exceeding the tolerance value and other nearby measurements to provide additional context. An event is created per smartphone when the car drives across a road pavement distress that causes sufficient vertical acceleration. The value of the tolerance is an estimation created by comparing the position of the created events to a manual tally of road pavement distresses.

The relationship between the events used in the case study and sensor standards is a follows: multiple events from a single real sensor, in this case, sensors in smartphones, make a RsDatastream entity. Multiple events with similar positions from sensors in different smartphones are gathered to form a VsDatastream entity.

The events from multiple smartphones can all describe a single road pavement distress. If this is the situation, then the positions of the measurements inside the event are also positioned near the road pavement distress location. The first step into creating a virtual sensor is selecting the raw data that falls into the chosen timeframe. Next, the centers of the events are calculated. The centers of events positioned near each other are grouped together based on the Density-Based Spatial Clustering of Applications with Noise clustering algorithm (Ester et al, 1996). The algorithm does not require information on the number of clusters to generate output, which is necessary as the number of distresses is unknown and changes through time. The position of the virtual sensor is created by taking the centers of all events in a single cluster and using them to calculate the average center. The vertical acceleration of smartphones in different cars is averaged and saved in the virtual sensor.

The accuracy of the virtual sensor position consists of two components: the deviation from the lane centerline and the longitudinal accuracy. During this case study, the data was gathered by driving on the right lane. The distance from the virtual sensor to the lane centerline is used to calculate the transversal precision. A clear relationship between the amount of raw data used and the transversal precision of the virtual sensors was visible; the more data is used to create a virtual sensor, the more precise its position is on the lane centerline. 1 shows the increase in precision when increasing the number of rides used as input to generate virtual sensors. A ride a single time series of data from start to destination gathered by 1 smartphone.

Nr. of rides	Nr of virtual sensors	Average distance to lane centreline (m)	Standard deviation (m)
10	27	1.54	0.96
20	26	0.98	0.92
30	32	0.82	0.66

Table 1. The transversal accuracy of the virtual sensors is dependant on the amount of rides or data used as input.

The longitudinal accuracy, described here as the accuracy along the length of a road section, could not be determined with a value. However, there is a clear correlation between the position of the virtual sensors and the locations of objects that are known to cause vertical accelerations like expansion joints.

A virtual sensor in itself does not support historical consistency. However, this shortcoming is circumvented by connecting the virtual sensors to assets which are historically consistent. In this case study, the virtual sensors are connected to road section geometry and hectometer post geometry, allowing contractors to monitor the increase of road pavement distresses through time per section or hectometer post.

5. Conclusion

The utilization of virtual sensors in combination with crowdsensing allows for the monitoring of static objects of interest with dynamic sensors like smartphones. Researchers can use virtual sensors to group the data gathered through crowdsensing and connect it to other objects, a downside is that the virtual sensor described in this paper does not maintain historical consistency. The virtual sensor allows the grouping and averaging of time series from different sources in both position and value. Simultaneously, the data structure of a virtual sensor is similar to the guidelines of the SensorThings API. While the sensor standard is currently not equipped to manage virtual sensors, the general structure of the standard can be expanded to cope with the new situations created by crowdsensing applications.

The case study illustrates a possible use of virtual sensors to connect data gathered by different smartphones in different cars to road pavement distresses and in turn to road sections and hectometer posts. The limitation of using virtual sensors in this scenario is that the averaging of the vertical acceleration is across different types of cars, meaning that the average value in the virtual sensor describes the vertical acceleration of the average type of car used while using crowdsensing.

References

Ester M, Kriegel HP, Sander J, Xu X (1996) A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise. In: KDD-96, AAAI Press, Munich, pp 226–231

Ganti R, Ye F, Lei H (2011) Mobile crowdsensing: Current state and future challenges. IEEE Communications Magazine 49(11):32–39, DOI 10.1109/MCOM.2011.6069707

Klunder G, van Aalst W, Willemsen D, van Vliet D, Leegwater G, Ruijs P (2010) Auto als Sensor Quick-scan naar detectiemogelijkheden van wegdekschade Inhoudsopgave. Tech. rep., TNO, Delft

Liang SH, Huang CY, Khalafbeig T (2016) OGC SensorThings API Part I:Sensing. OGC ® Implementation Standard

Wilson E (1997) Virtual Sensor Technology for Process Optimization Virtual Sensor (VS)