Incentivise Me: Smartphone-Based Mobility Detection for Pervasive Games

Thomas Tregel[1](✉), Felix Leber[2], and Stefan Göbel[1]

[1] Multimedia Communications Lab – KOM, TU Darmstadt,
Darmstadt, Germany
{thomas.tregel,stefan.goebel}@kom.tu-darmstadt.de
[2] TU Darmstadt, Darmstadt, Germany
felix.leber@stud.tu-darmstadt.de

Abstract. The ubiquity of smartphones with a plethora of sensors made it possible to develop context-based pervasive games playable by most people without any additional equipment required. However, popular games like Pokémon GO or Zombie, Run! show the minimal influence context has on the gameplay besides the user's position. Furthermore, in the context of movement and mobility the distinction between different modes of transport becomes necessary both for security and for environmental reasons.

In this paper we present a system that uses location information combined with public transport data to detect different types of mobility, especially vehicular mobility and uses this information for adaptation purposes in a prototypical location-based game. Our evaluation shows that our system can be used to reliably differentiate between public transport and driving a car within minutes and is able to adapt content in pervasive games according to the context surrounding the current user.

Keywords: Mobility · Mobility detection · Pervasive games · Context detection

1 Introduction

With the rise of smartphones, the demand for mobile games has also increased as a result of increased mobile Internet use. As a result, pervasive games that mix the real and fictional world have become more popular. In addition to larger screens, more and more sensors such as GPS receivers or magnetometers are being installed in smartphones. With these sensors it is possible to collect more data, so-called context information, about the user and his environment. One game that uses one of these data types is Pokémon GO, with which pervasive games have experienced a great hype. In 2016, Pokémon GO was the most widely used mobile game, which led to a positive change in behaviour for many users [1].

However, current pervasive games use little to no data about the players' movement, besides velocity threshold based approaches to detection potential vehicle usage. The aim of this paper is to detect the players' mode of transportation and individualize the content of pervasive games accordingly by collecting and analysing contextual

E. van der Spek et al. (Eds.): ICEC-JCSG 2019, LNCS 11863, pp. 470–476, 2019.
https://doi.org/10.1007/978-3-030-34644-7_48

information based on smartphones. This is especially important for serious games in the areas of eco-friendly behaviour.

2 Smartphone-Based Context Detection for Mobile Games

Contextual information related to pervasive games has been defined differently by several researchers. [2] and [3] present two well-known definitions for context with the latter being that that: “*Context is any information that can be used to characterize the situation of an entity”*.

Current pervasive games already include different context types exemplarily described in the scenario of Pokémon Go. Besides the player position as a main game input, the time context is used as a day-night cycle for both gameplay reasons and security concerns. Weather is detected via many different online services for the user’s position, for which open API can be used [4]. This weather information both influences game mechanics and notifies the user about upcoming weather changes.

Another information of particular interest with regard to pervasive games is the current activity of the user. This includes, walking, cycling, driving a car or the like. This is done by using sensor-fusion with e.g. the accelerometer, gyroscope, magnetometer, GPS, WiFi, pedometer and others [5]. Current games however only use a velocity-based static threshold to differentiate between valid movement and motorized travel.

The differentiation between certain activities is particularly difficult when sensor data hardly differs. In order to distinguish between different modes of vehicular transportation additional logic is required, which we will investigate in the next Section.

2.1 Differentiating Between Different Modes of Transportation

The detection of the player’s current activity plays an important role in context detection. Data when the user is in a vehicle is quite similar, which makes it difficult to tell whether the user is sitting in the bus or in his own car. We have examined multiple approaches regarding IMU-based modality detection in our previous work [6]. The downside here lies within the distinction between different vehicular modes of transportation, as accelerometer sensor data is influenced by many other aspects like road quality or vehicle suspension.

Enriching a vehicle discrimination by location data can be promising as shown by Thiagarajan et al. who have presented three distinguishing characteristics [7]:

- a car is significantly faster outside rush hours than a bus on its route.
- the mean distance between consecutive stops is significantly greater for cars during rush hour.
- drivers do not tend to stop at stops.

Stenneth et al. use routes of trains and bus stops to identify the means of transport [8]. In addition to that Montoya et al. also use timetables and live information of busses

[7, 9]. Location-based approaches need to incorporate mechanisms that handle variance in location sensor accuracy [10].

However, using position information can lead to significant errors or jumps, especially in large cities. Additionally, bus and train routes often overlap, so a lot of data is needed to distinguish between specific routes.

In Germany the availability of public transport data is very limited, due to the division into different transport associations, which usually do not provide open APIs. However, public transport stop information can usually be downloaded directly from the transport associations [11] or obtained via OpenStreetMap.

3 Detecting Public Transport and Game Adaptation

As investigated in [6] using a common API is reliable when detecting basic activities, excluding different modes of transportation. We have evaluated the Google Awareness Transition API [12], which detected the current activity (running, driving in a vehicle or standstill) in 45 tests (15 per activity) within a maximum of two minutes. Especially the change to running was very fast. It was also noticeable that it makes no difference which activity is changed from, but only to which. An overview of the results of the tests is given in Fig. 1.

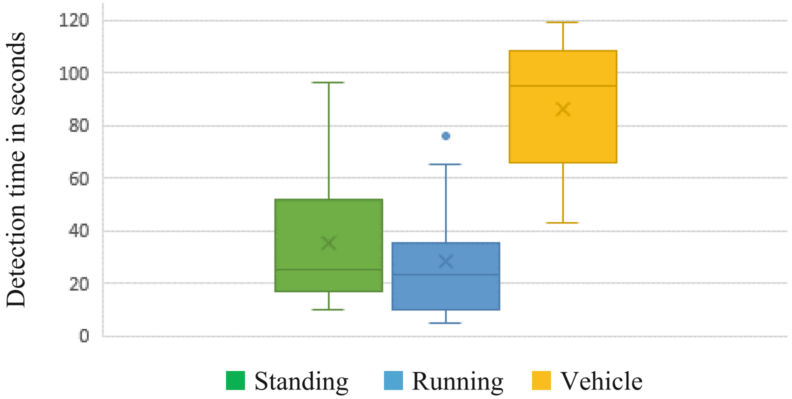

Fig. 1. Detection times of Google Transition API.

Thereby our approach starts as soon as the user is recognized by a common API as being in a vehicle. As noted in Sect. 2.1 it is important that all stops are detected. Therefore, we need to determine the minimum sensing frequency for location updates, as stops can be primarily be detected due to no change in player position.

For 30 journeys with bus and train, we have determined ten seconds to be the minimum downtime, which leads us to a required location update every five seconds according to the Nyquist-theorem. Due to higher basic update frequencies in location-based games, this could easily be increased without additional battery consumption.

At each position, route information is determined for the player's surrounding, which includes nearby routes and stops using our OpenStreetMap approach, which we extended by a public transport module. This information is then stored to reduce the number of consecutive network calls.

After obtaining all required information, we check to see whether a possible route is nearby. The procedure is as follows:

(1) A standstill is assumed at a speed of less than one meter per second. In this case, the determined route information is first searched for stops at a distance of δ = 50 meters (step 2). If the speed is higher, only nearby routes are searched (step 4).

(2) If a stop is found, it is saved. If there is the possibility to query departure times, the departures for all stops found will be retrieved (step 3). If no stops were found, the system also searches for nearby routes (step 4).

(3) If a departure is found within the time interval [−2 min, 10 min], the departure is also saved. Ten minutes were selected here due to the ten-minute guarantee of the RMV [14]. The two minutes were selected because the times in the timetable are often departure times, and therefore a short standing time is also taken into account. The values of the possible routes are then updated (step 5).

(4) The system searches for routes at a distance of δ = 50 meters around the current position. If one or more routes are found, they will be saved. If no routes are found within a time window of 2 min (represents 12 location updates with a frequency of 1 update per 10 s), the algorithm terminates and outputs the activity *Driving a car*. Then the values of the possible routes are updated (step 5).

(5) Each route that has been detected at least once in the vicinity is assigned a value. This value for a route r is calculated according to the following formula:

$$score_r = \frac{\omega_p * pos_r + \omega_r * stops_r + \omega_d * departures_r}{\#pos}$$

For this formula the weights $\omega_p = 1$, $\omega_r = 3$ and $\omega_d = 7$ have been chosen initially. pos_r represents the number of positions within the vicinity of route r. Additionally, $stops_r$ is the amount of correctly detected stops for route r and $departures_r$ the amount of valid departures within the given time window of step 3 for successfully detected stops. This sum is then divided by the number of positions processed so far, so that a value is generated as a function of time.

If a route's value succeeds a give threshold τ = 1 the value and route information of the highest scoring route is returned and the respective route is detected (tram, bus, train, etc.). In case all scores are below threshold τ the algorithm returns *Driving a car* (Fig. 2).

In addition, active Bluetooth connections and Wi-Fi information are analysed to influence the decision-making process. In case an active Bluetooth connection with a car is detected *Driving a car* is returned. Similarly, when the smartphone is connected to public transport Wi-Fi public transport becomes more likely.

3.1 Game Adaptation Based on Mode of Transportation

Game content adaptation for public transport is especially interesting as, given an attractive reward, these could change the player's mobility behaviour. These can be implemented using a quest-based system, similar to quests current location-based games already provide. One exemplary quest for public transport would be to *Use your local public transport for a full workweek in both directions (two times per day)*. As we further want to encourage users to use available public transport options we modify the allowed radius of interaction. This allows users when e.g. sitting in a tram to interact with content on a larger distance than when going by car. Because trams and buses do not take detours this is both required and convenient to allow enough content to be interacted with when going on a pre-determined route.

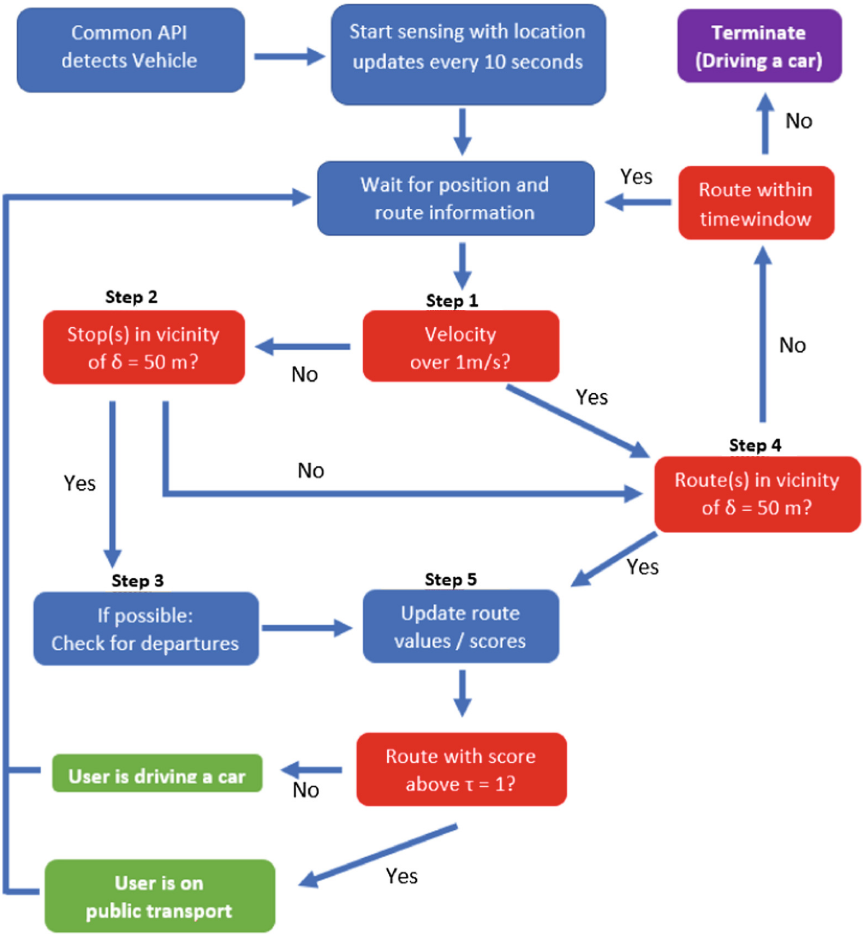
Common API detects Vehicle
Start sensing with location updates every 10 seconds
Terminate (Driving a car)
No
Wait for position and route information
Yes
Route within timewindow
Step 2
Stop(s) in vicinity of δ = 50 m?
Step 1
Velocity over 1m/s?
No
Yes
No
Yes
No
Step 4
Route(s) in vicinity of δ = 50 m?
Yes
Step 3
If possible: Check for departures
Step 5
Update route values / scores
User is driving a car
No
Route with score above τ = 1?
User is on public transport
Yes

Fig. 2. Public transport detection algorithm in a diagram.

4 Evaluation

In our evaluation we want to investigate the accuracy of our public transport detection approach and look at the feasibility and outcome of adaptation approaches for content selection and quest generation. The system was implemented within our GEOVis framework first presented in [13].

4.1 Public Transport Detection

In order to investigate the temporal aspect of the detection with regard to the means of transport, 20 tests were carried out, half in a car and half in a public transport. The aim was to determine with what accuracy the correct activity was detected and after what time it was valid. The results show that the correct activity or route is detected in 85% of cases during the journey. On average, the correct activity and route is detected after 179 s, having faster detection times for cars.

For 50% of journeys, the algorithm started detecting an incorrect activity or route during the journey before it was correctly detected. This happened on average after 65 s, which can be interpreted as a required time until a stable prediction can be done. It is also noticeable that at least one ‘wrong’ stop was detected in all car tests where a wrong route was detected during the journey (50%). This happened especially when stops were near traffic lights and the car had to stop. Due to the higher value of stops in the algorithm, this route was detected as wrong first. For this reason, the correct activity was not detected correctly during a journey. The public transport tests, where no correct route was detected, showed that the journey times were too short. Both consisted of only one stop and a maximum of two minutes of travel time. In both cases the correct activity, but not the correct route, was detected because several routes were driving this route.

Stops and departures of the specific routes were detected properly for real stops and departures which were within the defined time window.

5 Conclusion

In this paper we presented a concept for context-based adaptation in pervasive games focusing on differentiating between different modes of transportation, which is recognized by a combination of smartphone sensors and openly available APIs. An algorithm was designed to improve the recognition of the user’s activity, in particular the distinction between car and bus or train.

The in-game content is customized by player’s respective movement context, by both providing him quests that further encourage public transport usage and improve his interaction radius, that allows the access to more content in the vicinity. Our results show that the recognition can be improved with existing departure data of the public local traffic.

Acknowledgment. The research presented in this paper was partially funded by the LOEWE initiative (Hessen, Germany) within the research project, Infrastruktur – Design – Gesellschaft" as project mo.de.

LOEWE

Exzellente Forschung für
Hessens Zukunft

References

1. Althoff, T., White, R.W., Horvitz, E.: Influence of Pokémon Go on physical activity: study and implications. J. Med. Internet Res. **18**, e315 (2016)
2. Schilit, B., Adams, N., Want, R.: Context-aware computing applications. In: Proceedings of IEEE Workshop on Mobile Computing Systems and Applications (1994)
3. Abowd, Gregory D., Dey, Anind K., Brown, Peter J., Davies, N., Smith, M., Steggles, P.: Towards a better understanding of context and context-awareness. In: Gellersen, Hans-W. (ed.) HUC 1999. LNCS, vol. 1707, pp. 304–307. Springer, Heidelberg (1999). https://doi.org/10.1007/3-540-48157-5_29
4. De Pessemier, T., Dooms, S., Martens, L.: Context-aware recommendations through context and activity recognition in a mobile environment. Multimedia Tools Appl. **72**(3), 2925–2948 (2014)
5. Su, X., Tong, H., Ji, P.: Activity recognition with smartphone sensors. Tsinghua Sci. Technol. **19**(3), 235–249 (2014)
6. Tregel, T., Gilbert, A., Konrad, R., Schäfer, P., Göbel, S.: Examining approaches for mobility detection through smartphone sensors. In: Göbel, S., et al. (eds.) JCSG 2018. LNCS, vol. 11243, pp. 217–228. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-02762-9_22
7. Thiagarajan, A., Biagioni, J., Gerlich, T., Eriksson, J.: Cooperative transit tracking using smart-phones. In: Proceedings of the 8th ACM Conference on Embedded Networked Sensor Systems (2010)
8. Stenneth, L., Wolfson, O., Yu, P.S., Xu, B.: Transportation mode detection using mobile phones and GIS information. In: Proceedings of the 19th International Conference on Advances in Geographic Information Systems (2011)
9. Montoya, D., Abiteboul, S., Senellart, P.: Hup-me: inferring and reconciling a timeline of user activity from rich smartphone data. In: Proceedings of the 23rd SIGSPATIAL International Conference on Advances in Geographic Information Systems (2015)
10. Grimes, J.G.: Global positioning system standard positioning service performance standard. Washington, D.C., U.S. Department of Defense (2008)
11. Rhein-Main-Verkehrsverbund: RMV Open Data. https://opendata.rmv.de/site/start.html. Accessed 05 Mar 2019
12. Google: Google Awareness API (2016). https://developers.google.com/awareness/. Accessed 15 June 2018
13. Tregel, T., Raymann, L., Göbel, S., Steinmetz, R.: Geodata classification for automatic content creation in location-based games. In: Alcañiz, M., Göbel, S., Ma, M., Fradinho Oliveira, M., Baalsrud Hauge, J., Marsh, T. (eds.) JCSG 2017. LNCS, vol. 10622, pp. 212–223. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-70111-0_20
14. Rhein-Main-Verkehrsverbund: RMV-10-Minute-Guarantee. https://www.rmv.de/c/en/services/passenger-rights/rmv-10-minute-guarantee/. Accessed 05 Mar 2019