RoomService

a crowdsourced indoor positioning service

Sam Wong

sam.wong09@doc.ic.ac.uk

Department of Computing - Imperial College London

January 15, 2013

Contents

Abstract

Do this part at the end.

Chapter 1

Introduction

1.1 Motivation

The outdoor localization problem has been solved successfully using GPS. Its free to use nature has made many useful civilian applications possible, including Sat Navs and surveyors drones. The aim of this project is to provide a similar service for indoor localization. A free to use, open source service for crowdsourcing training data, and answering localization queries. The hope is this will spawn many new possibilities for indoor services, from one-click-pizza to novel location aware services.

1.2 Requirements

1.2.1 A easy to use java library for the Android platform

Android is the selected platform for this project. The main advantage is its popularity, open source nature, java backed system and less restrictive API.

This library should be easy to use and require minimum investment from users and developers.

1.2.2 Precision requirement

The precision level should aim to be high enough to be useful, but not too high such that the hardware requirements and computation cost becomes unjustifiable.

The current Wi-Fi localization provided by Android can already achieve a accuracy level of less than 50m.

These data come from Wi-Fi data collected from Google's mapping vehicles and companies like Skyhook[1].

This means built in Wi-Fi localization can already narrow down the results to a few buildings. To push the envelope, this project aim to provide room-level accuracy to application developers.

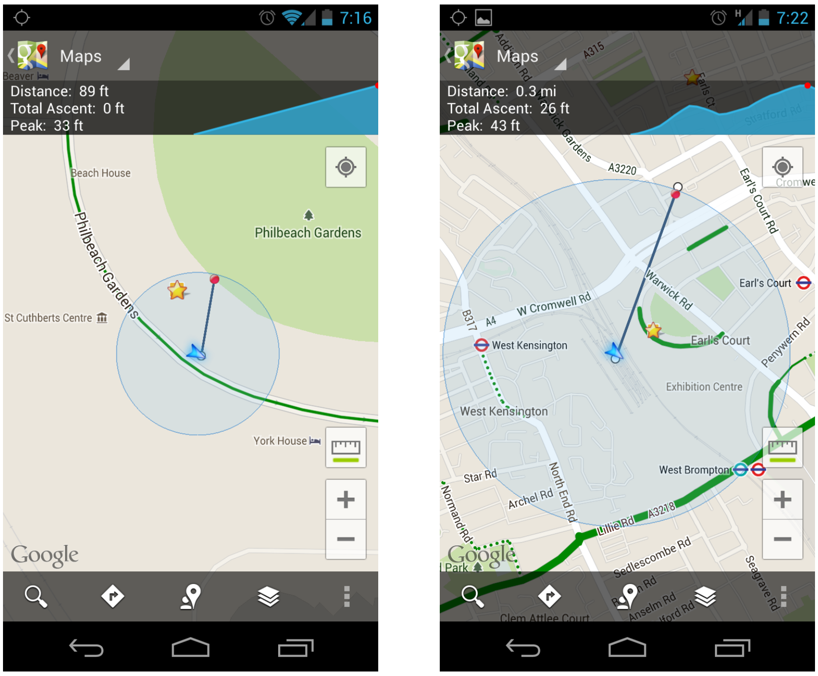
Room-level precision is chosen for this project as this can create many new possibilities for applications. For instance, a coffee shop owner may use this to automatically reward the most loyal customers.

1.2.3 Reliability, Scalability and Availability

To convince other developers to adopt my solution, it must have all these qualities. From my perspective, the system must be able to support the provisioned load, and make sure it is easy to scale up to deal with extra load.

From the other developers' perspective, this framework should never bring down their app or server, provide reasonable service in terms of latency and response time, and be highly availability such that they are comfortable to develop applications on top of it.

[1] http://www.skyhookwireless.com/



(a) Wi-Fi localization, Accuracy: 27m

(b) Cellular tower localization, Accuracy: 480m

Figure 1.1: Built-in localization accuracies. Star denotes the actual location

Chapter 2

Background

2.1 Android

Android is a mobile operating system based on linux. It is constantly evolving and new features are added to new API-levels. There are currently 17 API levels, where new API levels are only useable in later versions.

This is the most popular mobile operating system to date. Its market share is about 75% in the smartphone market, and about 45% in the tablet market.

Its popularity is a double edged sword, while it means the project could benefit more people, the heavy segmentation of the android market means there is no standard implementation. Currently more than 250 devices have been listed on the android website[1], the make and quality of the hardware varies, therefore no assumption can be made on the error bounds of the sensors.

The standard Android smartphone is equipped with microphone, speakers, camera, GPS, Wi-Fi, Bluetooth, accelerometers, gyroscope, and compass. Some high-end models feature more sensors such as light sensors and proximity sensors.

2.2 Indoor localization techniques

Indoor localization is still an open problem, here are some of the more successful methods.

2.2.1 Trilateration based techniques

Using satellites (GPS)

Each GPS satellite is equipped with an accurate atomic clock. Each broadcasted message from these satellites contains the timestamp of broadcast and the satellite position. A receiver can thus calculate how far the satellite is, defining a sphere.

When the receiver receives more than 4 satellite sources, the overlapping point of these spheres is the receiver's location.

As GPS requires a direct line of sight to operate, it doesn't work in indoors, and urban crayons. In the context of indoor localization, GPS is usually used to obtain the initial position. An example is to record which entrance the user is using, providing the users' initial position for indoor navigation.

Using beacons at known positions

Once the initial setup is done, localization can be performed using triangulation or trilateration.

This is one of the best performing approaches. However, this is only used in commercial settings as it has a high setup cost.

Beacons are typically implemented using bluetooth, selected for their high spatial selectivity[2], and low

[1] http://www.android.com/devices/
[2] Typical Bluetooth devices have a range of 5-30m

cost. Specialized beacons can be used to further improve accuracy. Such as spinning beacons[Chang et al, 2008].

Another approach is to install receivers at known locations to receive signals transmitted from transmitters attached to objects of interest. AT&T labs have used a ultrasound receiver network to locate objects and recover its orientation[Harter et al, 1999].

Using cellular towers

Cellular signal is radio wave, hence they can penetrate most urban structures. Many positioning software contains a database of cell tower locations, therefore a mobile phone can estimate its location with trilateration using signals from several cellular towers. However, as the cell phone have no way of knowing how much signal attenuation is caused by signal travelling through obstacles, the accuracy is measured in city blocks.

2.2.2 Fingerprinting

This approach usually requires no prior setup, nor knowledge of the building layout. Making it particularly scalable.

Wifi fingerprinting

This is a popular approach. The proliferation of Wi-Fi access points have made this a very viable option. Each wifi access point has a globally unique MAC address, providing a much smaller error bound compared to cellular tower.

If access point position is known, it is possible to perform trilateration as well. However, wifi signal can penetrate and reflect on walls. Hence the receiver cannot reliably measure the distance between access point and itself, therefore trilateration is effective in large open space environments only.

Magnetic field fingerprinting

This approach exploits the variation of background magnetic field to perform localization.

This variation comes from natural geomagnetic variations and man-made sources, such as electrical wirings and conductors. This variation provides a spatially variant and temporally stable fingerprint which can be exploited for localization.

There are several working demos on youtube[UNTNetworkSecurity Demo, IndoorAtlasLtd Demo], and many research papers on the topic.

With custom hardware, researchers have achieved accuracy within 1 meter 88% of the time[Chung et al, 2011].

Ambient Sound fingerprinting

This approach exploits the ambient sound caused by man made sources such as air conditioning and workstations. This is found to be spatially variant but temporally stable[Tarzia et al, 2011], and hence is exploited in localization.

The mathematical foundation on waves is strong and there are many efficient algorithms to perform analysis on frequency spectrum. Making sound waves suitable for use in mobile devices.

Tarzia et al have noted that sound based localization is less likely to confuse adjacent locations, and their implementation achieved a 69% accuracy, and have successfully distinguished pairs of adjacent rooms with 92% accuracy[Tarzia et al, 2011].

2.2.3 Hybrids and other techniques

Using Inertial Sensors

By double integrating acceleration, one can obtain displacement. However, in practice, the error accumulates very quickly. Hide et al have noted that "Low cost inertial sensors are often promoted as the solution to indoor navigation. However, in reality, the quality of the measurements is poor, and as a result, the sensors can only be used to navigate for a few seconds at a time before the drift becomes too large to be useful...In fact, even high quality tactical grade inertial sensors still experience drifts of hundreds of metres after periods as short as 5 to 10 minutes."

While they are not very effective on their own, Inertias Sensors have proved to be an important component in many successful indoor localization devices. Shoe-mounted Inertial sensors[Abdulrahim et al], Inertial sensors used in conjunction with computer vision[Hide et al, 2009], and Inertial sensors used in conjunction with map-matching techniques[Quddus et al, 2003] have reported positive results. These hybrid solutions obtains additional constraints from other sensors to greatly reduce the drift.

Using computer vision

Computer vision is typically used in conjunction with other techniques. One of the primary use is to introduce additional constraints to control drift. For example, it can be used to estimate attitude[Kessler et al, 2010], or estimate direction of travel[Hide et al, 2011].

In one implementation, researcher noted that man-made environment contains many straight and parallel lines in orthogonal directions. This is exploited to find the vanishing points in three dimensions to recover pitch, roll and heading. This is then used to correct the drift of INS. They noted that "Using the body solution, namely the camera and INS attached to a backpack, the vision-aiding yielded a 93% improvement in the heading error during evaluation tests. With a foot-mounted solution, namely the INS and camera attached to the ankle of the user, the horizontal position error decreased by 34 %."[Ruotsalainen et al, 2012]

Another approach is to use image bag-of-word. The state-of-art algorithm is robust and efficient[Csurka et al, 2004]. One obvious limitation is that the query must contain the features in the training data. This assumes the environment is relatively static, and the user will know the distinctive feature of the location (unless it is a video). So this will work better in a gallery than in a warehouse.

2.3 Evaluation

As this service is used mainly on mobile devices, there are a few limitations.

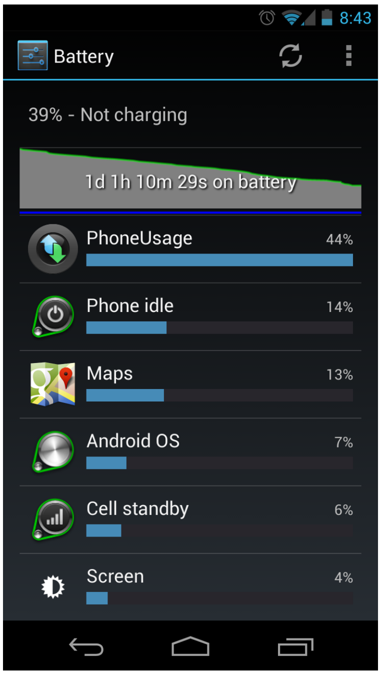
Firstly, the battery and bandwidth footprint must be minimized. Smartphone users have learnt to selectively switch off unnecessary sensors to reserve battery and data quota. Take GPS as an example, while it is very accurate, many people have it permanently switched off to preserve battery. This is because for casual navigation usage, WiFi localization is enough to help users locate themselves on the map. Moreover, Android provides a way of monitoring battery and data usage for individual apps. It is very common for people to complain about apps' excessive battery or data footprint. This leads to uninstalls and bad publicities. Application developers are very conscious of this problem, therefore the addition of RoomService must not significantly increase the footprint of their apps. Finally, if the footprint is small, then it would be possible to perform frequent location queries in the background, creating a responsive context aware experience for the end user. This means the service cannot be computationally intensive, and work should be delegated to the server as much as possible.

Secondly, the response time should be reasonable. Cellular tower and WiFi localization can be completed in seconds[3], therefore the response time of my service should be in the same order of magnitude.

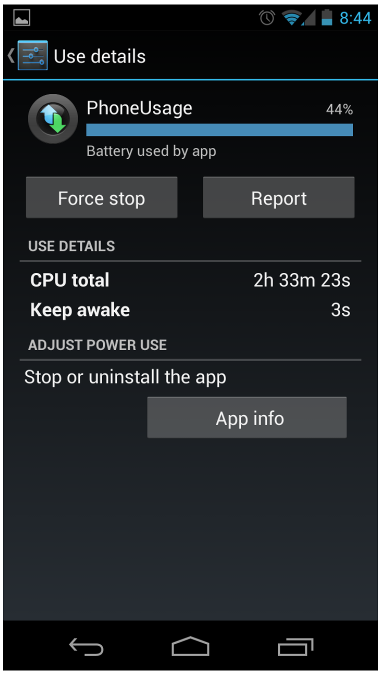
Thirdly, the service shouldn't require too much preparation for it to be useful. For example, requiring users to upload the floor plan of the building is too much work. It has been said that 1% of the users contribute almost all content, 9% of the users contribute a little, and the rest are lurkers[Jakob Nielsen, 2006]. Expecting there exists a 1% type of person in 90% of the buildings of interest is unrealistic.

With these points in mind, we can rule out computer vision related techniques. This is because vision software tends to be large in size, and computationally intensive. If this process is to be delegated to the server, then the image must first be uploaded to the server. An average image is about 2MB in size, which takes about 3 seconds to upload via 3G, or 1 second via Wi-Fi, and significantly longer for videos. While this is still acceptable in terms of time, the data usage is too high for frequent use.

[3]GPS typically takes more than half a minute to locate the user, and is therefore not included for comparison

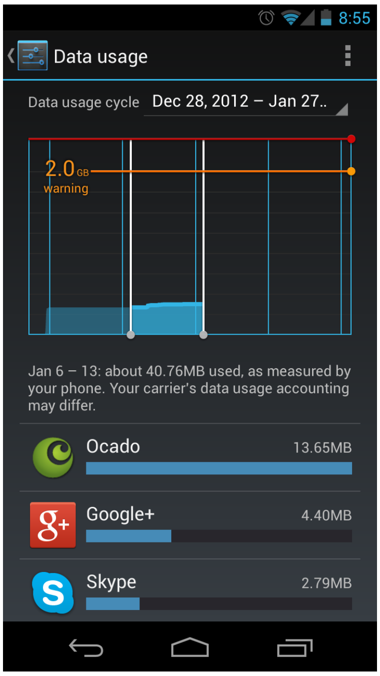


(a) Page showing battery usage of all applications

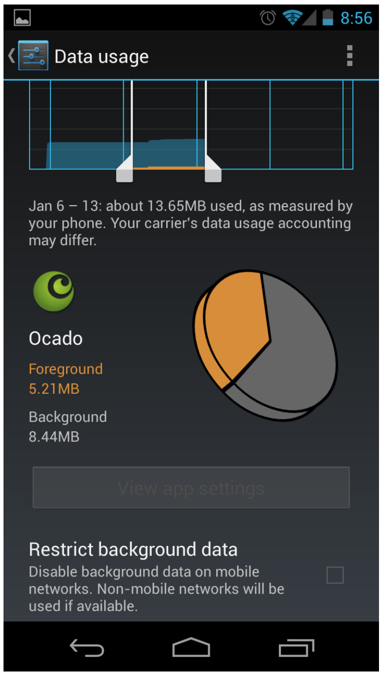


(b) Page showing the battery usage of an individual app

Figure 2.1: Android's built-in battery usage monitor



(c) Page showing data usage of all applications



(d) Page showing the data usage of an individual app

Figure 2.2: Android's built-in battery usage monitor

As the inertial sensors become a standard feature of smartphones, it is tempting to use it for navigation. However, this technique is only useful when integrated in conjunction with other techniques, such as vision and map-matching. Vision is deemed unsuitable for mobile devices, and map-matching requires too much investment from the users. Moreover, segmentation in device means every make of phone (if not every device) have sensors of different quality, and with their own sensor biases. Given the state-of-the-art is not suitable for navigation purpose, relying on inertial sensors alone is unlikely to produce acceptable result. Therefore this approach is also deemed unsuitable.

For trilateration based techniques, it requires a known beacon locations. The only viable options for mobile phone is to use cellular towers and Wi-Fi. Cellular tower based solution is already built in for android systems, and hence there is no point pursuing this route. To use Wi-Fi trilateration, one will need to build a database of Wi-Fi access point location. This again requires heavy user investment, and unscalable. Although this approach has a potential for high accuracy localization, the case to use this approach is not strong enough.

Out of all approaches outlined above, fingerprinting is the most suitable technique for the purpose. Preliminary experiment suggests that Magnetic field fingerprinting may not work in general settings, as the compass reading on my tablet varied greatly from point to point within the same room. Moreover, as magnetic field fingerprint is of low dimensionality, it is likely to collide in global scale.

Ambient Sound fingerprinting was considered, but to record audio samples is expected to raise privacy concerns. On the other hand, this is also a low dimensionality fingerprint, it is also likely to collide in global scale. While this technique may be added in the future to further improve accuracy, at this stage, Wi-Fi fingerprinting is the best way to go.

Wi-Fi fingerprinting has a lot of advantages. Firstly, it is extremely prevalent. In urban settings, it is very common to see a dozen access points. Each of these access points have a globally unique MAC address, so there is no chance of collision[4]. Secondly, there is no privacy concern, as MAC addresses and SSIDs are public information which the owners chose to broadcast. Thirdly, since we can combine both user supplied training data and the Android built-in Wi-Fi localization result, our result is expected to be strictly better than the built-in solution. Finally, a Wi-Fi fingerprint is easy to obtain, and very compact in size[5]. The load on CPU and bandwidth requirement are orders of magnitude lower than Ambient Sound fingerprinting.

Therefore, Wi-Fi fingerprinting is the technique of choice for the purpose.

[4] Unless they spoofed their MAC address, but this is rare.

[5] For example, data from a single access point can be encoded compactly in about 15 bytes. e.g. "68a86d0ce3a650", where 68:a8:6d:0c:e3:a6 is the MAC address, and 50 is the signal strength. Hence 1 KB can fit 70 access points.

Chapter 3

Project Plan

3.1 Prototyping

3.1.1 Create a proof of concept prototype

Done, the prototype is available for download at http://www.samwong.hk/RoomServiceClient.apk.

3.1.2 Improve the app to allow for easy data collection, performance testing, validation and stuff

This is going to take a few weeks.

3.1.3 Investigate if it is possible to share training data between devices

My experiment with Galaxy Nexus and Nexus 7 shows that training data from different devices are not compatible. The next step is to investigate the nature of the difference. Is it a constant bias, scaled bias, or just random?

Need to go out and collect loads of data to perform statistical analysis.

Best Case: There is a way of normalizing data. The normalizing factor can be mined from the training data.

Worse case: Every device has a separate set of data. Not the end of the world, but you will need to train every device you own, and retrain when you get a new device.

3.1.4 Create an intuitive design for contributing and managing training data

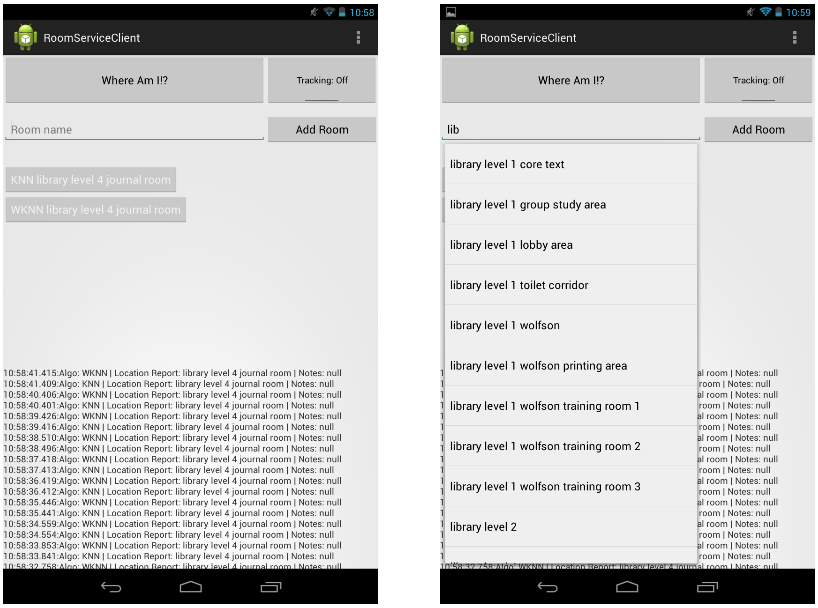
There are a few things desirable in the training data.

Firstly, data points from the same room should have very similar if not identical label. One way to do this is to offer suggestions. Is there a better way? How can we make sure training data from Huxley 311 and Blackett 311 don't both go to label "311"?

Secondly, the data should be mutable, so CoffeeShopA can be relabeled to McDonalds when the shop has changed hands.

Thirdly, the data should be easily separable by location, such that only a small relevant subset needs to be analyzed.

Lastly, how to structure this? Do we need to attach a global address to every label? e.g. "Room 123, Department of Computing, Imperial College London, Huxley Building, 180 Queen's Gate, London SW7 2BZ, UK"? Not undoable but is there a better way?



(a) After pressing 'Where am I?" (b) Label suggestion feature

Figure 3.1: Screenshots from the prototype

3.2 Cloudify

3.2.1 Evaluate different service providers

On Google as an app? On AWS as ubuntu server instances? Rackspace? Others?

Need to balance scalability, cost, availability and ease of use.

3.2.2 Evaluate what is the best design

Anywhere from one big machine with everything installed (say, mongoDB), to a bare bone server which talks to many dedicated service provider (say, DynamoDB).

3.2.3 Do it

3.3 Create documentation for the library

So people can understand how it works and use it!

3.4 Create some demo apps

3.4.1 A room level broadcast app

Just as Prof Guo visioned, a click and everyone in the room gets the message, video or file.

Chapter 4

Evaluation plan

For this to be a success, the performance should be at least as good as Ambient Sound fingerprinting technique. Ambient Sound fingerprinting is chosen as the baseline because this is the closest alternative viable option. I expect my solution to perform better in its plain form, i.e. only use training data created from the same device.

4.1 Tests planned

4.1.1 Well trained floor

The aim is to see how well it can tell where the user is in the best case. Aim to beat Ambient Sound fingerprinting.

4.1.2 Identifying untrained area

See if it can tell the user is in uncharted area, and prompt user to contribute training data. No baseline can be found in this area. Identifying a suitable metric to decide the 'well-trainedness' of the location will be a success.

4.2 Pass grade

Completing the full investigation on the concept should secure a pass grade (The Prototyping bit above)

4.3 Top grade

Completing the whole project plan, which includes a release-ready library, a live scalable service and a working demo (room level broadcaster) should secure a top grade.

Bibliography

[Chang et al, 2008] Chang, H.I., Tian, J.B., Lai, T.T., Chu, H.H., Huang, P.: Spinning beacons for precise indoor localization. In: Proceedings of the Sixth ACM conference on Embedded network sensor systems, SenSys (2008)

[Harter et al, 1999] Andy Harter, Andy Hopper, Pete Steggles, Andy Ward, Paul Webster. The Anatomy of a Context-Aware Application. Proceedings of the Fifth Annual ACM/IEEE International Conference on Mobile Computing and Networking, MOBICOM'99, Seattle, Washington, USA, August 1999, pp. 59-68.

[UNTNetworkSecurity Demo] Magnetic Field Based Indoor Localization, by UNT Network Security Lab. http://youtu.be/iHVoI1n89TY

[IndoorAtlasLtd Demo] IndoorAtlas' indoor location technology. IndoorAtlasLtd. http://youtu.be/PkehW3fkpLQ

[Chung et al, 2011] J. Chung, M. Donahoe, C. Schmandt, I.-J. Kim, P. Razavai, and M. Wiseman. Indoor location sensing using geo-magnetism. In MobiSys '11, 2011.

[Tarzia et al, 2011] S. P. Tarzia, P. A. Dinda, R. P. Dick, and G. Memik. Indoor localization without infrastructure using the acoustic background spectrum. In MobiSys '11, 2011.

[Hide et al, 2010] C. Hide, T. Botterill, and M. Andreotti, "Vision-aided IMU for handheld pedestrian navigation," in Proc. Institute of Navigation GNSS Conference, Fort Worth, Texas, 2010, pp. 1-9

[Abdulrahim et al] Abdulrahim, Khairi, et al. "Low Cost, High Accuracy Positioning for Indoor Navigation."

[Hide et al, 2009] C. Hide, T. Botterill, and M. Andreotti, "An integrated IMU, GNSS and image recognition sensor for pedestrian navigation." In Proc. Institute of Navigation GNSS 2009 Conference. Fort Worth, Texas

[Quddus et al, 2003] M. Quddus, W. Ochieng, L. Zhao, and R. Noland.A general map matching algorithm for transport telematics applications. GPS Solutions Journal, 7(3):157-167, 2003

[Csurka et al, 2004] Csurka, Gabriella, et al. "Visual categorization with bags of keypoints." Workshop on statistical learning in computer vision, ECCV. Vol. 1. 2004.

[Kessler et al, 2010] Kessler, Christoph, et al. "Vision-based attitude estimation for indoor navigation using vanishing points and lines." Position Location and Navigation Symposium (PLANS), 2010 IEEE/ION. IEEE, 2010.

[Hide et al, 2011] Hide, Chris, Terry Moore, and Tom Botterill. "Low cost vision aided IMU for pedestrian navigation." J. Glob. Position. Syst 10 (2011): 3-10.

[Ruotsalainen et al, 2012] Ruotsalainen, Laura, Jared Bancroft, and Gerard Lachapelle. "Mitigation of Attitude and Gyro Errors through Vision Aiding." International Conference on Indoor Positioning and Indoor Navigation. Vol. 13. 2012.

[Jakob Nielsen, 2006] Jakob Nielsen. "Participation Inequality: Encouraging More Users to Contribute" Jakob Nielsen's Alertbox: October 9, 2006. http://www.nngroup.com/articles/participation-inequality/