Abstract

The Internet has become a place to share experiences. Millions of pictures and videos are shared every day on social networks. But the growing market of smartphones and smartwatches is quickly shifting the paradigm for consumer media usage and impressing the need for more mobile-centric solutions. In this work, a social network prototype, called FlyrAR, was developed and designed to be used outdoors, with augmented reality features through sensor fusion methods, using the technology already embedded on daily mobile devices. FlyrAR allows the user to leave a virtual mark on the location he/she took a photo, and others can find the content by pointing the camera of the smartphone near the location. For a complete solution, it was implemented a back-end using Google App Engine with two Cloud Endpoint clients Android and Web. The prototype was optimized to avoid too much loading from the cloud and content on the screen by comparison the postal-code of the current location of the user with the geotag of the content, and then filtered from distances between 2 to 20 meters. Such application will open new doors for focused advertising and understanding outdoor user behavior.

Keywords: Augmented reality, social network, sensor fusion, Android, Google App Engine

1. Introduction

More than one third of the global population has a smartphone [1]. Low-cost smartphones are opening new doors for marketing and e-commerce in new markets where the consumers didn’t have Internet. However, on well established markets, smartphones are changing very quickly the paradigm of the consumer, emerging a need for mobile-centric approaches. With the growing number of social networks, the consumer becomes active on creating content, instead of just getting information from the usual media formats (magazines, television and radio).

Nowadays, social networks like Facebook trace the interactions of the user with the content, and online habits to create a commercial profile to be used as advertisement tool [2]. However, this data is very limited because external conditions and the environment motivates online behavior. But there are applications that rely on location of the user, like Foursquare, that allows the users to leave a virtual "check-in" on the places they visited using their location. Comparative to Facebook where the user is using passively the location, on Foursquare they are active on creating geo-based content to interact with other users, allowing the generation of more accurate profile for advertisement. There are other application areas to use geolocation, like geography, where the data is used to improve land fields, using photographs that complement satellite images from Google Earth for 2 [3] and 3 dimensions [4]. On a public service side, there is a Portuguese application called NaMinhaRua [5] that allows the user to leave comments and virtual pictures in places that need improvements (street holes, garbage, vandalism), which information is then retrieve to relevant entities. Closer to the final consumer, other solutions try to embedded augmented reality for art [6], games and culture. But, since these applications are only used for the purposes of the app, they don’t get general social behavior data. So, these geolocations features are not completed explored for final consumer in this new paradigm.

There is a need for a solution that focused on a user more mobile-centric, so the data is used in a less intrusive way but more productive. In this project, it was developed a social network prototype for Android smartphones to be used outdoors. The core of the application is allowing the user to take a picture with text on a specific location and leave a virtual mark on that place. Then, other users can see the content on that location by pointing the camera to see it in augmented reality. This application uses the sensors of the device - location (Wi-Fi + cell towers), accelerometer, gyroscope, magnetometer and camera - by applying sensor fusion methods and tridimensional illusion techniques to make possible the natural interaction as augmented
reality. Since the goal was to develop the tools for a social network, this application was integrated on GAE (Google App Engine) with one more client - Web application - and a simple Android Wear application for notifications purpose.

2. Background
The majority of the smartphones, not counting with low-cost and designed for elder people and children, have at least a camera, a location system, 3-axis accelerometer, 3-axis gyroscope and a 3-axis magnetometer.

Smartphones’ location services are normally simplified as GPS (Global Positioning System) but it is more complex than that. Current smartphones and other mobile devices use what is called a Hybrid Location System that combines information from GPS, Wi-Fi signal and cell towers to retrieve an accurate location. GPS is the most accurate but only works outdoors, it takes more time to update and it consumes a lot of battery; Wi-Fi and cell towers are much faster to update and they work indoors and outdoors, but they have less accuracy.

A gyroscope (or gyro) measures angular velocity, that in know conditions, it allows to get the orientation of the smartphone [7, 8]. However, errors are being integrated as well, which causes a drift on the signal. So, gyro cannot be used to estimate an absolute orientation. An accelerometer and a magnetometer measure the gravitational and magnetic field of the Earth, respectively, and in theory, together allow the calculation of a absolute orientation. However, both are subject to noise [9, 10]. By combining the data from accelerometer and gyro is possible to estimate the orientation relative to the gravity. Although for many applications is enough [11, 12, 13, 14], for augmented reality is not. By adding the data from the magnetometer to the other sensors, it is possible to measure a complete orientation relatively to the gravity and magnetic field of the Earth. The motion tracking of the smartphone is measured by considering three orientation angles: azimuth, pitch and roll. The smartphone coordinate system is illustrated on the Fig.1 with the origin located on the center of the screen.

By combining the information from the three sensors is possible to estimate the current orientation through the implementation of filters. The most common is the Kalman filter [15]. In fact, its implementation shows accurate and efficient results [13, 16, 17, 18] but also complications if used in this project. The iterations for linear regression needs sample frequencies that exceed the band of the motion of the user that varies from 512 Hz [19] to 30 kHz [20].

Online applications that often has new content and users interacting with each other needs a database to save and retrieve content. Cloud computing allows the access through different platforms (desktop, smartphone), different users, scalability depending on demand, and monetization by usage metrics like time and type of content. Such services can be divided in three categories:

- Software-as-a-Service - designed to be used by final consumers using the web (e.g. Dropbox and Zendesk);
- Platform-as-a-Service - set of tools and services that allows the developer to make new applications in a fast and efficient way (e.g. Google App Engine and Microsoft Azure Services);
- Infrastructure-as-a-Service - hardware and software like servers, network, operative systems (e.g. Amazon Web Services and Rackspace).

3. Implementation
3.1. Architecture
FlyrAR was developed with a client-server architecture with GAE as back-end for the three interfaces Android mobile, Android Wear and Web. Each component has its own functions:

- Data back-end - to store flyrs and profiles;
- Smartphone - to create flyrs and to display them in a map or with augmented reality;
- Smartwatch - to display notifications about new flyrs;
- Web - to create flyrs and to display them in a map;

On Fig.2 is illustrated the communication between the back-end, the two Cloud Endpoint clients and the smartwatch.
The mobile app and the Web app communicates directly with GAE through a localhost, so they are called Cloud Endpoint clients. From the functional point-of-view, the core of the system is on the mobile app. Using the smartphone, the user can create a flyr, by taking a picture with text on it, which is going to be saved on the database along with the coordinates latitude and longitude, and a postal-code tag. The content can then be accessed by other users, using the smartphone’s camera to find it in augmented reality, or on the map using the mobile app or the Web app. Every time news flyrs are available on the database, the smartphone send a notification for the smartwatch with the number of flyrs created. The Web app is more limited in sensors data than the mobile app, so it can only display a World map for the user interact to see and create flyrs without visual content (used only for debugging purposes).

3.2. Database
In this project, GAE was used as real-time database. For the data to be processed by multiple platforms it was developed an API with permissions to Android mobile and Web. These clients must be able to get and create specific information, identified with its own key, like user profiles and flyrs. These objects are called entities, and each of them has its own properties. An entity user has the properties: unique ID and the name; and an entity flyr has the properties: unique ID, text, picture, latitude, longitude and postal code. Since the flyrs are always associated to a user, a third entity was created to generate a unique key for each user based on the Google e-mail account. Every time a flyr is created it has the user as its parent.

It was implemented 8 methods for the Cloud Endpoint, so the clients could execute an action on the back-end, like create, load and save information: save profile, load profile, update profile, delete profile, create flyr, list flyrs, update flyr and delete flyr. On these methods, it was used Objectify (or Ofy()), which is an API specially designed to communicate with GAE [21]. By considering the property postal code on each flyr, it is possible to load the flyrs that have equal postal code compared with the current user location, avoiding excess loading from the database.

3.3. Smartphone application
In order to interact with the flyrs with augmented reality to look like they belong to the real World, first it was developed a sensor fusion algorithm to get a 3 dimensional orientation of the smartphone, and then an algorithm to find the flyrs based on the location.

3.3.1 Orientation algorithm
The implemented method to get absolute orientation of the smartphone is called sensor fusion [22]. This method was developed considering that the accelerometer and the magnetometer gave an absolute orientation without drift over time, but they have noise and low time response when analysed in shorts periods of time. On the other hand, the gyro gives rotation data with good accuracy at high frequencies, but with small errors accumulated over time, that damages the estimation the absolute orientation. To compensate that, the gyro data is filtered with a high-pass filter, and the low frequency data is replaced with the data from the other two sensors which was applied a low-pass filter. This configuration, illustrated on the Fig.3, is called complementary filter [23] and it was based on the proposal of Shane Colton [24].

![Figure 3: Schematic of data fusion sensors with a complementary filter. It has this name because the components of the filter contributes with low and high frequencies, respectively, to get an absolute orientation with high accuracy.](image-url)
variable \textit{gyroOrnt} of the Equation 1 that has the absolute orientation based on the gyro.

\[ \textit{gyroOrnt} = \textit{gyroOrnt} + d\text{Time} \cdot \textit{gyroRot} \quad (1) \]

where \textit{dTime} (seconds) is the time since the last update of \textit{gyroOrnt} and the current value, and \textit{gyroRot} has the values of the angular velocity of the gyro. But, since the information of the absolute orientations is derived from the measures of the gyro, the drift still remains and it needs to be removed [25]. On each low-pass filter update of the orientation based on accelerometer and magnetometer, the old values and the new ones are weighted with a coefficient. This new value is saved on the variable \textit{accMagOrient} on the Equation 2.

\[ \textit{accMagOrnt} = a \cdot \textit{accMagOrnt} + (1 - a) \cdot \textit{newAccMag} \quad (2) \]

where the \textit{newAccMag} has the new value from the accelerometer and magnetometer. The term \((1 - a) \cdot \textit{newAccMag}\) can be seen as the low-pass filter part that lead to slow changes. This calculation slowly changes the absolute value of the orientation, minimizing the errors of the new measures. As a result, the reaction time of both sensors increases. The term \(a \cdot \textit{accMagOrnt}\) also can be seen as the high-pass component of the filter. But in the Equation 2 there is no high frequencies because each new value was low-pass filtered. In this way, that part is replaced by the weighted result that came from the integration of the gyro after high-pass filtering at Equation 1. That will cancel te constant changes of the orientation based on the gyro, including its drift. The final operation of the filter is given by the Equation 3.

\[ \textit{ornt} = a \cdot \textit{gyroOrnt} + (1 - a) \cdot \textit{newAccMag} \quad (3) \]

The variable \textit{ornt} is a 3-value vector with the angles azimuth, pitch and roll. Following the API documentation from Android for sensor fusion, the value used for \(a\) should be 0.80 [26], but since for bigger values, the update changes are smoother, the value used was 0.95 which didn’t compromised slow or fast variations of the angles while rotating the smartphone.

\[ x = \Delta \lambda \cdot \cos \phi_m \quad (4a) \]
\[ y = \Delta \phi \quad (4b) \]
\[ d = R \cdot \sqrt{x^2 + y^2} \quad (4c) \]

where in Equation 4a, \(\Delta \lambda\) is the difference between longitudes in radians, and \(\phi_m\) is the mean of the latitudes in radians; in Equation 4b, \(\Delta \phi\) is the difference between latitudes in radians; and in Equation 4c, \(R\) is the radius of the Earth that is approximately 6371000 meters. Notice that in this approximation the Earth is round, which is completely acceptable for small distances.

Now that we have the distance, its missing the location of the flyr in degrees on the absolute orientation. Latitude and longitude angles increase to North and East, respectively. Since the absolute orientation is known, and also the latitude and longitudes of both user and flyr, it is possible to know in which quadrant the flyr is, relative to the user (see Fig.4). This method allows the estimation of the flyr inside a smaller space.

3.3.2 Augmented reality algorithm

Having the orientation defined, the distance and the location of the flyr relative to the user must be estimated. The augmented reality scheme is applied to the user that wants to see the content. The way it is done is by turning on the camera. At this point the location of the user is static, so as the flyrs around, which makes possible to calculate the distances. In this project, it is only possible to see flyrs in augmented reality between 2 to 20 meters of the user. The location of the user and the flyrs are based on the latitude and longitude angles using a equi-rectangular approximation [27] given by the Equations 4.

\[ x = \Delta \lambda \cdot \cos \phi_m \quad (4a) \]
\[ y = \Delta \phi \quad (4b) \]
\[ d = R \cdot \sqrt{x^2 + y^2} \quad (4c) \]

For sake of simplification, let’s consider that latitude and longitude have equal contribution for the calculation of the distance between the user and the flyr. If the latitude and longitude of the flyr is subtracted to the latitude and longitude of the user, the result is a variation, in degrees, from the flyr to the user. Let’s consider that the flyr is located between 0 degrees and 90 degrees (at 1st quadrant). If the variation on latitude is divided by the sum
of the variation on latitude with the variation on longitude, the result is the relative contribution of the latitude for the calculation of the distance. This contribution can be placed on the green line illustrated on the Fig.5 in which 0 degrees is when there is only contribution of the longitude, and 90 degrees when is only contribution of the latitude. By drawing a line (in red) from the origin, this line will intersect the point where the flyr is located (in blue) and the arc that has the absolute orientation of the camera.

Figure 5: Representation of the method to get the absolute angle of orientation, based on the coordinates of the flyr relative to the user.

The orientation in which the camera must point to see the flyr is then given by the Equation 5.

\[
\text{flyrOrnt} = \text{minAngleQuad} + \frac{\Delta \text{lat}}{\Delta \text{lat} + \Delta \text{lon}} \cdot 90
\]  

(5)

where \(\text{flyrOrnt}\) is the absolute orientation in degrees where the flyr is and \(\text{minAngleQuad}\) is the smallest angle of the quadrant - on the 1st quadrant is 0, on the 2nd is 90, and so on.

Now let’s add to the flyrs the illusion of presence on a real space. Starting with size on the screen, it must be inversely proportional to its distance. For example, if at 2 meters the flyrs has 2 cm in width on the smartphones’ screen, this means the size decreases with distance at a constant equal to \(2 \cdot 0.2 = 0.4\). The calculation of the size for any distance is given by the Equation 6.

\[
\text{flyrSize} = \frac{\text{distMin} \cdot \text{tamMax}}{\text{currentDist}}
\]  

(6)

In this example, a flyr at 20 meters has a size of 2 mm on the screen. Other important feature for the illusion is the motion of the flyr on the screen on the opposite side of the rotation of the smartphone to look like it is static in space (see Fig.6).

Figure 6: Motion of the flyr on the screen on the opposite side of rotation of the smartphone to make the illusion of the flyr being static on a tridimensional space.

This principle must be applied for either azimuth (left and right) and roll (up and down) motions of the smartphone. For the pitch (rotating the smartphone) only rotates the flyr on the screen. But the illusion of distance is not complete. At close distances, the horizon seems its narrowing and going up, because we have height relative to the ground. This means flyrs at longer distance must have some elevation on the screen (see Fig.7).

Figure 7: Schematic of the top-view of the FOV of the user [left] and (same) screen view on the smartphone [right].

The last thing that is missing is 3D effects and shadow. When it is nigh, the shadow is not needed, but during the day it is. There is 4 main possible shadows (see Fig.8) that are dependent on the location of the user, the position relative the flyr and the time of the day.

Figure 8: Possible variations of shadow during the day.

3.4. Web and smartwatch applications
Like other social networks, a full application must have a mobile version and a web version. Normally, the web version has equal or less features than the mobile. In this project, the Web application was developed only with the features of the seeing the map.
with flyrs, and create flyrs by clicking on the map just for simulation purposes. When the user creates a flyr on the map, its longitude, latitude and postal code are automatically saved on the database. Since the computer may not have location system, the current location of the user must be placed manually.

By the time this project was purposed, smartwatches were just an entry technology without big demand. But its growth in the market lead mobile application to expand its capabilities to these new platforms. In order to make this project updated with this new technology trend, it was developed a simple communication between the smartphone and a smartwatch for fast and easy notifications access. This happens when the map of the smartphone or Web is populated with new flyrs, sending a notification to the smartwatch with the number of new flyrs created.

4. Evaluation
FlyrAR application was tested by 4 people, in different locations - Oporto (Portugal), Lisbon (Portugal) and Boston (United States) - and with different Android devices - Moto G2, Samsung S5 and LG G3 - to study the interaction and functionality. These devices are compatible because they have the three inertial sensors needed, location service, camera and an Android version bigger than 10. The tests made were the following:

- Interface;
- Exact location of the user;
- Creation of flyrs with text and picture;
- Communication between back-end and Cloud Endpoint clients;
- Visualization of the flyrs on the maps (mobile and Web);
- Visualization in augmented reality;
- Smartwatch notification.

During the installing of the app, the user is notified to allow the app to use the camera, location and Internet. After the installation, the icon appears on the launcher of the smartphone with the logo and the name “Flyr”. By clicking on it, the app starts with a 1.5 seconds screensplash and a notification to turn on the location service if the user didn’t do it (see Fig. 9).

4.1. Flyr creation
To create a new flyr, the user clicks on the black bottom located on the right corner which is going to open the camera to take a picture (see Fig.11). After the picture is taken, the user can add a text to it.

Figure 9: Screensplash [left] and notification to turn on the location service [right].

When the location is obtained, the main layout appears with a map centered on the current location of the user. This layout is where the user has all the important shortcuts and notifications, but the most important: the map to see where he/she is and what is surround him/her (see Fig.10). The testers quickly understood the interface.

Figure 10: Main layout with the map covering almost all the screen. It is centered on the user current location. No flyrs exist around the user.

Figure 11: Camera opened to take a picture [left] and adding a text after the picture was taken [right].

After the flyr is created, it is saved on the database, and the map is updated with the new flyr (see Fig.12). By clicking on it, it is possible to see the text and the name of the creator.

Figure 12: Main layout with the map covering almost all the screen. It is centered on the user current location. No flyrs exist around the user.
Using the Google API Explorer website through a localhost, it was created more flyr around the current location. This interface allowed the testing of the API methods created for the Cloud Endpoint. It was generated 4 more flyrs. By updating the map on the smartphone, the new ones appeared (see Fig.13).

When the map on the smartphone was updated, a notification was sent to the smartwatch to tell the user that there were new flyrs in the area (see Fig.14).

Also, the Web app was updated too (see Fig.15).

4.2. Flyr visualization

Then it was tested the augmented reality feature. The current flyrs were not between the distances 2 and 20 meters relative to the user, so it was created 3 more flyrs closer, but now using the Web application. These flyrs were saved on the back-end and updated on the other Cloud Endpoint client (the smartphone). By turning on the camera of the smartphone, it was possible to see the new flyrs in augmented reality (see Fig.16).

Notice that when rotating the device to the right, the closest flyr disappeared, and the smaller ones moved to the left on the screen, giving the illusion that the flyrs were actually static on 3D space (see Fig.17).

Also, the flyrs located far away from 20 meters didn’t appear because they were filtered by distance. The ones in augmented reality are clickables.
so the user can click on it to see the content - picture and text (see Fig. 18).

Figure 18: Content of the clicked flyr - a picture of a dog with the text "Check this out ahah".

5. Conclusions
The following project had the goal to develop a functional prototype of a social network with augmented reality features, by using location sensors, inertial sensors and the camera of the smartphone. The use of so many sensors may lead to a big battery consumption, so an optimized approach was followed to obtain accurate and efficient results. The sensor that drains more battery on the smartphone is GPS. In this project, the location was made using Wi-Fi combined with cell towers signals. By doing that, we took the risk of getting location errors closed to 5 meters, but for the purpose of this application it does matter, because what is important is the context where the flyr is located. Also, these services are only called when the user opens the application or when he/she turns on the camera, otherwise the location service is off. In fact, using the application all day didn’t influence the battery usage compared with a normal day use of the smartphone. In theory, the second thing that drains a lot of battery is the inertial sensors: this application uses all three. To avoid excess of battery usage, it was implemented a sensor fusion method that uses low frequency data processing (250 Hz) which is all it is needed to capture human motion when using the camera as augmented reality tool. This method gave very accurate results for the three orientation angles: azimuth, pitch and roll in landscape mode. When the smartphone was static, the angles pitch and roll only showed variations of 1 degree, but the azimuth varied 2 degree. To improve that, a simple filter with 5 sequential weighted samples was used. The camera also drains considerable battery. The only optimization possible was on the flyr itself. When the user takes a picture, the quality is reduced to 480x320 to avoid excess of data on the database and when loading from it. Other types of optimizations for the functionality of the application were the use of the postal code tag as a filter, and then the calculation of the distances to get flyrs from 2 to 20 meters by using equirectangular approximation. On the first case, it was used the postal code because the only data that it available for comparison is geographic. Choosing the name of street for comparison may be too limitative, and choosing the country may be too extreme to load so many content from the cloud (if any). Also, the user probably won’t travel far away distances to see the content. Then, in augmented reality, to estimate the distance between the user and the flyrs, it was used a metric that has only 2 squares and one root, which allows fast calculations with good accuracy for small distances. It was rejected flyrs closest than 2 meters to avoid flyr occupying to much area on the screen, or more than 20 meters because of the obstacles on the city (people, buildings). By using the GAE as back-end allowed an intuitive and real-time integration with Android and Web app, which in future development of other clients, like iOS, the same Cloud Endpoint can be used.

On the business side, FlyrAR is the World’s first application that has outdoors socialization in real-time that can be mixed with spatiotemporal analysis. If the app is setup to use the location services during all the interaction, this data can be saved anonymously to train large scale machine learning models to understand user outdoors behaviour. For example, it is possible to learn the human motion patterns that depend on external factors like the time of the day, year, or weather conditions, and internal factors like gender, age and nationality of the user. This allows more strategic and focused geolocalized advertisement. Also, the flyrs were developed in the way that it is easy to customize like change colors, text, geometric form, and motions that can be used to get user attention.

In summary, it was developed a complete solution for mobile, desktop and wearable devices to understand the viability of a new approach as a social and business tool, showing very promising results.

References
Crowdsourcing Geographic Knowledge. 2013, 139-159.


