Smartphone Internal Sensor-based Offline Displacement Estimation and An iOS App Development

James Taylor and Qingzhong Liu
{jamestaylor@shsu.edu; liu@shsu.edu}

Department of Computer Science, Sam Houston State University, Texas, USA 77341

Abstract. GPS, cell towers, Wi-Fi access points and beacons each can be used to determine location of a cellphone. However, each is an external piece of hardware to the cellphone, and each has limitations to the quality of service provided. For example, GPS signals can be compromised by a number of events - including solar activity, man-made interference and malicious faking of GPS signals [1]. As an alternative, an app can be made using only sensors internal to the phone. This paper details our iOS app for offline displacement estimation by only using common cellphone sensors such as the accelerometer and magnetometer, without any WiFi and GPS signals, to estimate a user's location as they walk to any desired location. This research demonstrates the use of a modern approach to dead reckoning. The results show that the error in estimation using this approach can vary from 6cm to 15cm for every meter walked, with increasing inaccuracy for more intense activities.

Keywords: Dead Reckoning; GPS; Magnetometer; iOS, iPhone.

1 Introduction

The way that maps are made and used is constantly evolving. From cave paintings [2], to online maps and from looking at the stars to looking at a phone that receives communication from satellites orbiting the earth, the history of displaying location remains fluid.

Cellphones can determine their own location via satellites, Wi-Fi signals and beacons. However there are pros and cons to each one of these broadcasting methods. For example, if a person is indoors, the received GPS signal is degraded or lost, leading to inaccurate readings or no reading at all. This research the creation and analysis of an app that can be used on a smartphone to determine a person's location that travels from a known point to an unknown point without communicating from any external manmade device.

This paper is organized as follows: In Section 2 we analyze the current methods that are used to determine location with external devices. In Section 3 we discuss how to determine location without an external device. Section 4 reviews the methods used to build the app. Section 5 presents and analyzes the data collected from the app. Section 6 presents a list of potential changes to improve the app and contains concluding remarks.

MOBIMEDIA 2018, June 21-22, Qingdao, People's Republic of China
Copyright © 2018 ACM
DOI 10.4108/eai.21-6-2018.2276641

2 Background

There are several different methods to determine a position on a map. Each method uses a single or multiple manmade devices to transmit data.

2.1 Global Positioning System

The Global Positioning System (GPS) is owned by the United States Government and was developed by the United States Department of Defense. It is currently maintained by the United States Air Force [3]. There are other positioning systems such as GLONASS and BeiDou that have been developed by other countries [4], but GPS currently is the most commonly used system and is available worldwide for free. GPS is comprised of 31 satellites orbiting the earth and continually transmitting data. This data contains an identifier for the satellite, the time as determined by an on-board atomic clock, and ephemeris data (the orbital information of the satellite) [5]. Devices on Earth use data from at least 4 satellites concurrently to determine its position and altitude. Locating position using GPS is referred to as trilateration.

The satellites orbit in a manner that at any point on earth surface at least 4 satellites are visible. However, obstructions such as a roof, shield or natural barrier can degrade the received transmission. In addition, defects, solar activity, and the malicious faking of GPS signals can compromise the data received

2.2 Cell Towers and WiFi

Cell towers can help cellphones approximate their location. This is accomplished by the cellphone detecting the signal strength of one or more cell towers in a method called triangulation. As the number of cell towers in range of the cellphone increase, the location accuracy also increases. Wi-Fi access points can also be used to determine location by using a database comprised of correlating mobile device GPS location data with associated MAC addresses.

The FCC reports that when cell tower triangulation is used, a phone’s location can be determined to within 3/4 square mile [6]. Thus, this type of positioning is best for non-precise measurements. This accuracy decreases in more rural areas as the number of cell towers is lower. Urban areas can experience a decrease in accuracy due to obstructions blocking and reflecting the signal. If no cell towers are within range, this method cannot be used. Similarly, the Wi-Fi location estimations cannot be used if there are no Wi-Fi access points. Both cell towers and Wi-Fi access points are non-operable if a power outage were to occur.

2.3 Beacons

Beacons are small devices that can communicate via Bluetooth Low Energy, which has a short range [7]. They are built using Apple’s iBeacon protocol, and are used for sending small, context specific information to users. For example in a retail space, coupons may be sent, or estimated wait time can be sent [8]. Beacon proximity is measured as immediate, near and far. If the iBeacon is transmitting latitude and longitude information to the device location is determined via triangulation.

3 Deterimining Location without External Signals/Devices

The previous technologies share a common problem: without them a cellphone can no longer determine its location. The following are methods a phone can use to find a change in location with its own sensors.

Cellphones have a growing number of sensors built in that help them make sense of their surroundings. The two most helpful for this application are the magnetometer and accelerometer.

A magnetometer measures the earth's magnetic field. The data from a magnetometer is communicated via three properties, X, Y and Z, representing magnetic field readings in the left/right, forward/backwards and up/down axes. The values of these properties can be used to simulate a compass and thus point to magnetic north. This magnetic data can be accessed using the CMMotionManager class [9] in an iOS app. Apple has an API that will simulate a compass that can be accessed by instantiating the CLLocationManager class [10], such that the magnetometer data is processed and filtered to display magnetic north. The compass heading (direction) is updated several times a second. Heading is defined as direction the device is pointed. For simplicity we decided to forego using the raw data directly, and have used the CLLocationManager to calculate the headings.

The accelerometer measures gravity. It is helpful in determining the movement. Using measurements from the accelerometer a pedometer can be made. Apple makes such data readily available via the CMPedometer [11]. Apple does not specify how steps are measured, but we assume it is using the accelerometer. Newer iPhones read and process motion data such as the accelerometer in a low-energy, always-on coprocessor. Compared to the accelerometer, the using GPS on a phone has a power cost of 100x [12]. The pedometer data is updated in intervals of about 2-5 seconds. In addition to tracking steps, it is possible to determine velocity, but using low-cost sensors (like the one in the iPhone) is "very poor and [is] simply unusable" [13].

Using both of these sensors can allow for a type of positioning calculation known as dead reckoning. Dead reckoning has been used for boats and airplanes to determine location using heading and speed [14]. When an initial location, heading and speed are known, a navigator can estimate where they will be after any given period of time. The accuracy of this method is dependent on the accuracy of the sensors used to calculate the displacement.

4 Displament Estimation and iOS App Development

This app is built for iOS 10 and was tested on an iPhone 6 Plus. It functions by having the user type in their latitude and longitude. We recommend that each coordinate has six degrees of precision. We used http://www.latlong.net to find determine the current location, which accesses Google Maps and is built using a Google Map API. The user should zoom as far as possible, and then select their location. The GPS coordinates are displayed at six degrees of precision. The user then presses the "start sensors" button to get both the CLLocationManager and CMPedometer to begin alerting the app for value updates. When the user presses "start estimation", the app will begin estimating the user's new location using the real-time data from the aforementioned objects. The location estimation is displayed as two labels which represent the latitude and longitude coordinates.

The heading is displayed relative to true north using real-time results from the CLLocationManager. A compass (represented by the magnetometer) is able to read magnetic north. Magnetic north is a point on the Earth where the northern lines of attraction enter the Earth [15]. However, maps are usually made using true north instead of magnetic north. True north is the point on the earth where longitudinal lines converge. It is represented as 0º latitude and 0º longitude. The difference between true north and magnetic north is called the magnetic declination. Magnetic declination will change depending on where a person is on Earth and when the measurement was taken.

iOS will automatically convert magnetic north to true north if it has a network connection at any point. When a user presses the "start sensors" button in this app, a network connection is needed for a few moments to determine magnetic declination, and then can be turned off prior to estimating the change in location. We adjusted the heading readings in increments of 90º based on the orientation of the phone, so that the user can hold the phone as they please throughout the app lifecycle.

The heading is updated several times every second. If the current heading were to be used at the moment the pedometer update occurred, much information would be discarded and a less accurate result would occur. For example, if a person were to make a right turn the moment before the pedometer updates, the displacement along the prior angle would not be calculated. To account for the change in direction each heading update is averaged at the time the pedometer updates.

To find the average heading direction, the Cartesian coordinates of the heading are found in relation to a unit circle. These can be found by taking the cosine of the magnetic heading for the x axis and the sine of the magnetic heading for the y axis. These coordinates are averaged and then placed in the two argument arctangent function:

$$\alpha = \arctan 2(y, x) \tag{1}$$

The result is the average heading. It is necessary to use this function because a simple summation and average of heading can often lead to incorrect results. For example, if a person were to walk at 355º for one minute, and then at 5º for one minute, an average of those two headings would incorrectly be calculated as 180º which is the opposite direction.

The average heading is both calculated and reset upon each pedometer update. The app allows the user to also calculate the average heading independent of the distance traveled by pressing the "find average button", but this must be done before estimating the new location begins.

CMPedometer has a property called distance which estimates the distance in meters traveled by the user. We chose not to use this because how this number is computed is not exposed in the documentation, and thus may decrease the accuracy of the app. Instead we placed the average walking stride, 0.57m, into the code. The distance is calculated every time the CMPedometer notifies the app that the step count has changed by multiplying the step count by 57cm. The app also allows for power-walking and jogging activities which were calculated at 1.03m and 1.32m per stride, respectively.

With the initial location, average heading and distance traveled, there is enough information to estimate the user's location as represented by latitude and longitude. To represent location, latitude and longitude are utilized. Latitude is a measurement that indicates how far north or south a point is on the earth. The range is from 0º around the equator to +/-90º at the north and south poles, respectively. Each degree represents 110.574 km irrespective to location on Earth. Longitude is a measurement that indicates how far west or east a point is on the Earth. The range is from 0º at the Prime Meridian (Greenwich) and +/- 180º on the opposite side of the planet. Unlike latitude, the distance between each longitudinal degree differs based on the point of reference. At the north pole north and south poles, where all longitudinal lines converge, there is no distance between each degree. At the equator, the distance between each longitudinal degree is roughly 111km. Thus, the equation to find the longitude *lot* is

$$lot = 111.32km * \cos(\theta) \qquad (2)$$

where θ represents the latitude in radians.

To find a new latitude and longitude, the first step this app takes is to read the average heading and calculate the x and y components on a unit circle. This is done using a switch statement that switches on intervals of 90º. The latitudinal displacement d(y) is found by multiplying the x component by the distance *s* traveled

$$d(y) = \sin(\theta) * s \qquad (3)$$

The longitudinal displacement d(x) is found as follows

$$d(x) = cos(\theta) * s \qquad (4)$$

The resulting latitude *lat(1)* is found by dividing the vertical displacement by 110574m:

$$lat(1) = lat\,(0)+\mathrm{d(y)/110574m} \qquad (5)$$

The new longitude *lot(1)* is found dividing the horizontal displacement by 111320m times the cosine of the initial latitude in radians

$$lot(1)=lot(0)+\mathrm{d(x)/(111320m*cos(lat(0))} \qquad (6)$$

The timing of these calculations is important. The least frequently updated sensor, the step counter, is be the period at which the new location would be calculated. Upon the step count update, the average heading is found and the new latitude and longitude are calculated. The new latitude and longitude then become the initial latitude and longitude for the next location update.

5 Experiments and Simulation Analysis

To test this app, we mimicked real life movement, such as following a windy route instead of walking in directions in intervals of 90º. To demonstrate the accuracy of the average heading algorithm we choose routes that would otherwise produce inaccurate average headings. The routes had to be of a meaningful distance and one that would take between 30 seconds and 5 minutes. Thus, routes of 100m, 200m and 400m were selected. Because inaccuracy would result between each trial, 20 trials for each experiment were conducted so that trends could be identified. This data would also help determine how to improve the app for future releases.

The first route is pictured in Figure 1. It begins at 33.466661º, -111.915158º and ends due north east at 33.467156º, -111.914765º. The total distance is 102m. The results of the 20 trials of the first experiment are pictured in Figure 2. Most of the results ended up to the west and slightly south of the actual endpoint, with three outliers to the east. The average latitude and longitude of these results is 33.467156º, -111.914765º, which is 14m from the actual endpoint. This translates into ~14cm error introduced for every 1m traveled.

The second route (Figure 3) also starts at 33.466661º, -111.915158º but ends due west at 33.466566º, -111.916188º. The total distance is 198m. The results of the 20 trials of the first experiment are pictured in Figure 4. All of the results were found to be north east of the actual efigurend point, with one outlier due east in the skate park. The average latitude and longitude of these results is 33.466789º, -111.916024º, which is 29m from the actual endpoint. This translates into ~15cm error introduced for every 1m traveled.

The third route (Figure 5) starts at 33.473952º, -111.919274º and ends at the same point. It follows the second lane in a track, which has a distance of 407m. The results of the 20 trials of the first experiment are pictured in Figure 6. Most of the results were found to be north east of the actual end point. The average latitude and longitude of these results is 33.474176º, -111.919234º, which is 25m from the actual endpoint. This translates into ~6cm error introduced for every 1m traveled.

The fourth route (Figure 7) starts at 32.295327º, -111.222369 and ends at the same point. It follows a lap around a driveway which has a distance of 251m. This route was also used for the fifth and sixth experiments. The average result for the fourth experiment was 32.295653º, -111.222857º. In the fourth experiment the results were somewhat clustered North West of the end point with a few outliers. The average result for the fifth experiment was 32.295424, -111.223280º. The fifth experiment did not exhibit any clustering and produced results mostly due west of the end point. The average result for the sixth experiment was 32.295879º, -111.222528º. The sixth experiment had a clustering north west of the end point and a few outliers.

The sixth experiment was used to test a mixture of activity types and screen orientations. In this experiment jogging started at the first, then at point 1 on Figure 7 walking took place. At point 2 a pausing for 10 seconds occurred and then switched to power-walking. At point 3 walking assumed again. On the fifth step of each activity type we rotated the phone by 90º to the right in order to initiate an orientation change in the user interface.

All results are summarized in Table 1. The range of error measured for the walking experiments is between 6cm/m and 15cm/m. The range of error for the power-walking, jogging and mixed activities was 23cm/m and 34cm/m. The majority of the walking data collected is clustered together, meaning that although somewhat inaccurate, there is a high level of consistency. The most inaccurate data came from the jogging data in experiment 5. The third experiment had a high degree of accuracy and less than half the error compared to the first two experiments. The suspected reason for this accuracy is because the route ended at the same point as it started, and the change in direction was relatively uniform.

The first factor that led to inaccuracy in the results is the compass estimated an accuracy of 10º-15º. These results seem to indicate this is a conservative estimate. The second is the step distance. It is unlikely that each step in the 14km walked to be equidistant. In every calculation of a new location, some of the positional data are lost. For example, if a person walks in an 10m S-shaped path, the linear distance from start to end would be <10m. The app will still calculate the change in distance as 10m in that direction. The error increases with every result. The error increases the less linear the path is. It is near impossible to hold the phone exactly perpendicular to the body, because of which the phone will not always be pointed in the direction traveled. In each case the phone was placed about 5cm-10cm perpendicular to the sternum with right hand. We noticed that even when walking in a straight line the reading the compass would change slightly with the slight movements of the tester's body. Accuracy decreased for power-walking and jogging activities. This is most likely attributed to the larger variance in step distance. When changing the orientation of the phone there are a few heading measurement that are inaccurate until following measurements are adjusted by the heading manager. There may have been a few instances where the initial point was reset in experiments 4 and 6.

The data used to calculate distance and location had 2 and 6 degrees of precision respectively. Over large distances, this can result in distorted results as rounding errors and uncollected data accrue.

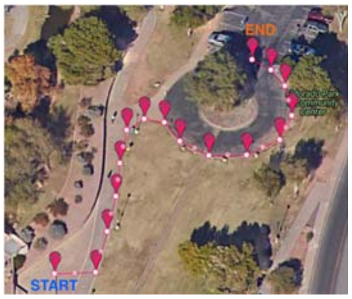

Fig. 1. route for experiment

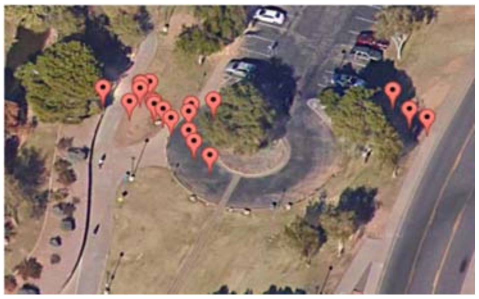
Fig. 2. Endpoint estimations for experiment 1

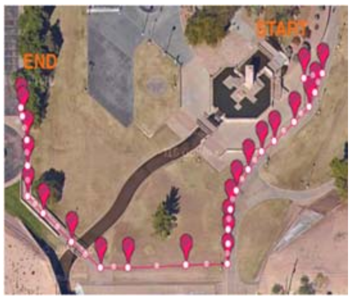

Fig. 3. Route for experiment 2

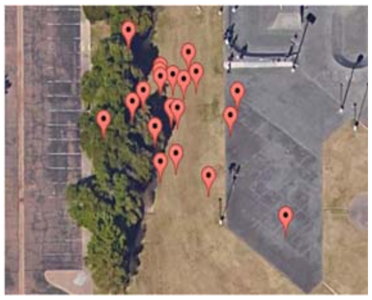
Fig. 4. Endpoint estimations for experiment 2

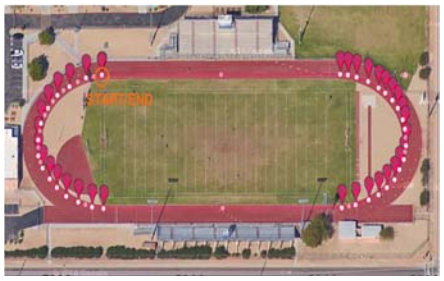

Fig. 5. Route for experiment 3

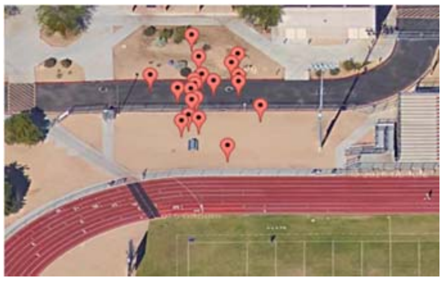
Fig. 6. Endpoint estimations for experiment 3

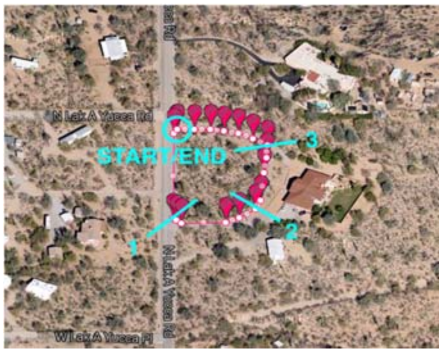

Fig. 7. Route for Experiments 4, 5 and 6 with markings for activity changes in experiment 6

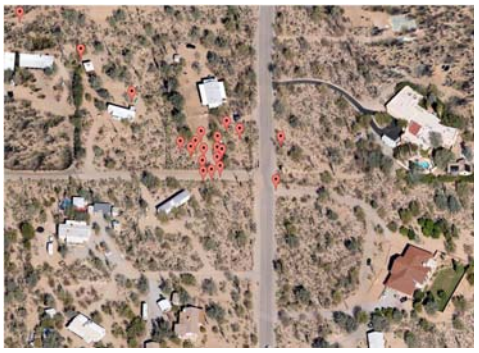

Fig. 8. Endpoint estimations for experiment 4

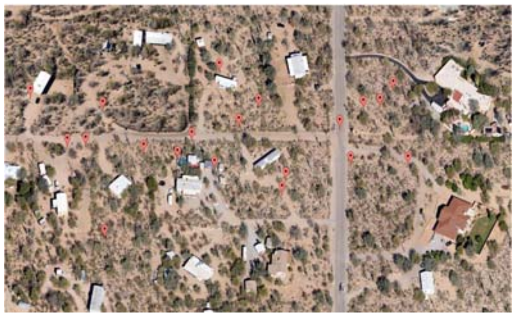

Fig. 9. Endpoint estimations for experiment 5

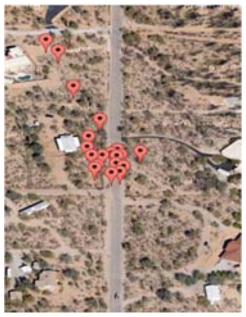

Fig. 10. Endpoint estimations for experiment 6

Table 1. Details and results of each experiment.

Experiments	Activity	Distance (m)	Error (m)	Error (cm/m)
1	Walk	102	14	0.14
2	Walk	198	29	0.15
3	Walk	407	25	0.06
4	Power-walk	251	58	0.23
5	Jog	251	86	0.34
6	Mixed	251	63	0.25

6 Conclusion and Future Work

Every method of determining location includes a list of pros and cons. The same con that GPS, beacons, cell towers and Wi-Fi access points have is their reliance on external manmade devices. As the number of devices increases, the number of points of failure also increases. This modern implementation of dead reckoning is suitable for use cases that can handle between 6cm to 15cm amounts of error per 1m traveled. The three largest limiting factors of this technology are 1) the accuracy of the sensors in the iPhone, 2) human error in holding the phone without erroneous heading readings and 3) the increased variability of strides for more intense activities. The first item may be remedied by the phone manufacturer in future phone releases assuming they place improved sensors in the device. The second item cannot be overcome without perhaps a harness. The third item may need to rely on a different method to estimate stride distance or pace. Finally, the code can also potentially be optimized to achieve a better walking stride estimation and non-linear route estimations. Similar to the other methods of determining a location, the app presented in this research document does have a level of accuracy sufficient to handle a number of use cases.

The scope of these future changes is the same as the initial app: determine the location of a person without an external manmade device. The following are items that can worked on for future iterations:

1) Have a locally stored map that the user can access to pinpoint their current location instead of needing to connect to the internet.

2) Have a locally stored and developer-accessible database that will contain the necessary data to calculate the magnetic declination instead of relying on network services.

3) Improve step distance estimation. While calculating displacement only using the accelerometer is very difficult, it is possible that some machine learning could help determine whether a stride is below or above the average stride. Because the CMPedometer includes a property that estimates distance, this could be experimented with to see if it is more accurate.

4) Increase the frequency with which average headings are calculated in order that a more accurate route can be estimated. For example, if 4 steps are reported by the pedometer, the array of headings can be split into four segments, and the average of each segment can each be assigned to one step. This would allow the data to more accurately fit the non-linear paths a user would walk.

Acknowledgments. The support for this study from Sam Houston State University ORSP office is highly appreciated.

References

1. Spirent, *Magnetic Declination* (2016) [Online]. Available: https://www.spirent.com/-/media/White-Papers/Positioning/Fundamentals-of-GPS-Threats.pdf
2. BBC News, *Ice Age Star Map Discovered* (2000) [Online]. Available: http://news.bbc.co.uk/2/hi/science/nature/871930.stm
3. NASA, *Global Positioning System History* (2015) [Online]. Available: http://www.nasa.gov/directorates/heo/scan/communications/policy/GPS_History.html
4. kuschk, *A Look at GPS Alternatives* (2012). [Online]. Available: http://basementgeographer.com/a-look-at-gps-alternatives/
5. P. Dana, B. Penrod, *The Role of GPS in Precise Time and Frequency Dissemination. (July/August 1990) GPS World. Available: http*://ilrs.gsfc.nasa.gov/docs/timing/gpsrole.pdf
6. M. Tran, *Cost Effective Location Detection Techniques Used by the 911 Help SMS App to Overcome Smartphone Flaws and GPS Discrepancies* (2015). [Online]. https://transition.fcc.gov/pshs/911/Apps%20Wrkshp%202015/911_Help_SMS_WhitePaper0515.pdf
7. CSR, *Bluetooth® Low Energy* (unknown). [Online] Available: http://www.bluetooth.org%2FDocMan%2Fhandlers%2FDownloadDoc.ashx%3Fdoc_id%3D227336&usg=AFQjCNGdNYpGP1eaaHGLFvFmDjQ9tPtaNw&sig2=Q_AKh8XkLfuTC4tileaSsA

8. A. Bleau, *Over 100 Use Cases and Examples for iBeacon Technology (2015).[Online].* http://blog.mowowstudios.com/2015/02/100-use-cases-examples-ibeacon-technology/
9. Apple, *CMMotionManager* (2016). [Online] https://developer.apple.com/reference/coremotion/cmmotionmanager
10. Apple, *CMMotionManager* (2016). [Online] https://developer.apple.com/reference/corelocation/cllocationmanager
11. Apple, *CMPedometer* (2016). [Online] https://developer.apple.com/reference/coremotion/cmpedometer
12. A. Kandangath, G. Badie, *What's New in Core Motion* (2015). [Online] Available: https://developer.apple.com/videos/play/wwdc2015/705/
13. CH Robotics, *Using Accelerometers to Estimate Position and Velocity* (2016). [Online]. Available: http://www.chrobotics.com/library/accel-position-velocity
14. N. Bowditch, *The American Practical Navigator: An Epitome of Navigation (2002).* http://msi.nga.mil/MSISiteContent/StaticFiles/NAV_PUBS/APN/Chapt-07.pdf
15. GIS Geography, *Magnetic North vs Geographic (True) North.* (2016) [Online] Available: *http*://gisgeography.com/magnetic-north-vs-geographic-true-pole/