VOL. 10, NO. 15, AUGUST 2015 ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences

www.arpnjournals.com

DEVELOPMENT AND TESTING OF AUGMENTED REALITY APPLICATION FOR SMART DEVICES BASED ON EMBEDDED SENSORS DATA PROCESSING

Ondrej Bilek[1], Ondrej Krejcar[1], Ali Selamat[1, 2], Reza Mashinchi[2] and Kamil Kuca[1]
[1]University of Hradec Kralove, FIM, Center for Basic and Applied Research, Rokitanskeho 62, Czech Republic
[2]Universiti Teknologi Malaysia, Faculty of Computing, Johor Baharu, Johor, Malaysia
E-Mail: Ondrej.Bilek@uhk.cz

ABSTRACT

Augmented reality is nowadays well known solution to present to users a comfort of real world view with a virtual one. We can found many examples of more or less specific solutions but most of them are focused to use an embedded sensors of modern mobile devices, where the developed application are used. Our paper would like to describe some basic info about possible platforms for application development as well as to introduce several issues to lead you when designing an application for. Most important issues are described with examples of specific procedures and algorithm samples of programming sensors on currently the most widely used mobile platform Google Android. Case study is presented as an example of photo application which is based on augmented reality elements.

Keywords: sensor, mobile, embedded, augmented.

INTRODUCTION

As power and possibilities of mobile devices rapidly grown with massive extension of smart devices such as smartphones or tablet computers, many sensors were integrated to extend possibilities and mainly, to help users with common use of smart devices [1-6].

Smart devices also integrate functions of some other devices [7-9]. Few years ago users need phone for calling or texting, camera for taking photos, MP3 player for listening music, GPS navigation or map to planning their trips [8, 9, 10], computer to opening documents etc. Nowadays, all these functions can be handled with single device, which offer great computational power with many possibilities in body of mobile phone or tablet computer. A brief comparison of supported and mandatory sensors is presented inside this paper on one case study example of augmented reality [11] application.

While several existing platforms on which the mobile smart devices are based, there is a one leading – Google Android [6, 7, 8, 12] which is described in this paper in more detail, as well as conditions for developers to use these embedded sensors are outlined. Google Android is the most used mobile platforms of these days and it's still rapidly growing and expanding to other markets. Another great benefit of developing app for Google Android is multiplatform developer tools, relatively cheap app publishing on platform store and much information around web.

After theoretic part of a paper, the practical parts follows. First parts describes proposal of sample application and second describes implementation, functions and testing of sample application. Sample application uses principles of augmented reality, where view from device camera is used to include data from sensors, what can be enhanced to defect management system [13] or some kind of advisory solution [14-16].

Developing Mobile Application using Embedded Sensors

The following section describes the basic features of the development environment (Software Development Kit - SDK) for major mobile platforms, which are summarized at (Table 1). Currently we can cover these ones: Google Android, Apple iOS, BlackBerry and Microsoft Windows Phone. This section describes the current situation in the late of 2013 and takes into account the current versions of operating systems, which are Android 4.2, iOS 6, BlackBerry 10 and Windows Phone 8. More information about comparison of platforms for mobile application development as well as embed sensors fulfillment of all platforms can be found [17].

Proposal of Sample Application

Functions of sample application was defined as follows:
Show data from GPS and sensors of environment (temperature, humidity, atmospheric pressure) Show tilt of device towards water level and direction from north Implement simple GPS navigation, this navigation should show distance and direction to coordinates, which should by user enterable Application should run on majority part of existing smart devices with Google Android operating system [1, 2, 6, 7, 12]

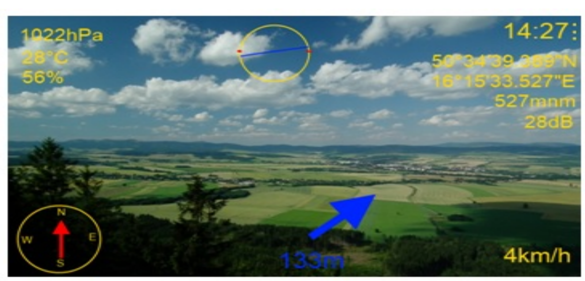

Figure-1. Design proposal of GUI for sample application.

ARPN Journal of Engineering and Applied Sciences

Sample application is using principles of augmented reality. Basic stone of application is real-time view from device's back camera. This view is enriched by overlay with sensor data. Firstly we focused on practical part in the sense of GUI design prior to final definition of UML and all functions as in classical way of application design. For this reason we are concerned on proposal of application. Preview of application, which you can see on (Figure-1), was created and functions of future final application was set.

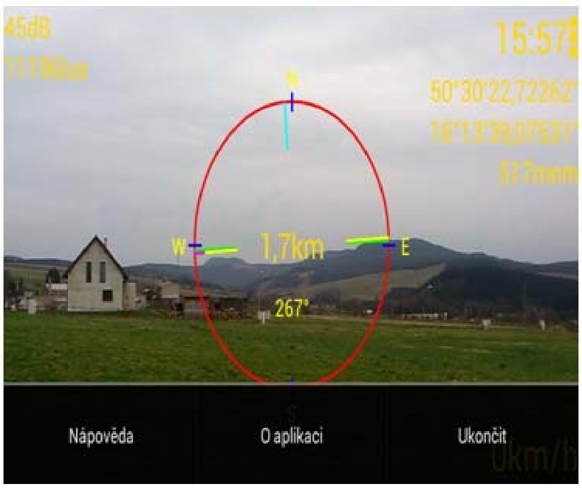

Figure-2. Final GUI of developed augmented reality application.

Important Parameters of Developed Solution

Application is used to add sensor information to camera view. We can see various sensor information on sides of camera view and multifunctional circle indicator at middle of view, which contain:

Compass - yellow indicator, magenta end towards to north

Spirit level - green indicator

Azimuth - yellow text, degrees from magnetic north

Navigator - distance at middle of circle, cyan indicator show direction (Figure-2). Application could be used for simple GPS navigation. Some text fields are also clickable (Figure-3):

Time shows main menu

GPS longitude shows actual GPS coordinates in other formats

SET or distance open coordinate settings dialog. For coordinates on west or south hemisphere, use - before degrees.

Atmospheric pressure open reference pressure dialog (in hPa). This pressure is used to calculate altitude from barometer. If 0 is set, standard pressure is used.

Figure-3. Final GUI of developed augmented reality application – location parameters definition.

Implementation of Augmented Reality Application

This chapter describes implementation of final version of sample application for Augmented Reality. Design and structure of application was modified against proposal.

Selected Algorithms for Sensors Implementation

Developed solution cover operating with several important sensors using interesting algorithm like: Coordinates saving, converting and validation; Tilt calculation towards water level; Azimuth east from north calculation; and Calculation of direction to coordinates. As example we will describe several of these.

Tilt processing

The tilt calculation is based on the modified procedure published in [18]. The first step of the calculation is a transfer of acceleration vector (array of values g) from the accelerometer to normalized ones. This is done by dividing of individual components by the square root of the sum of the components squares:

```
float n=(float)Math.sqrt(g[0]*g[0]+g[1]*g[1]+g[2]*g[2]);
g[0] = g[0]/n;    g[1] = g[1]/n;    g[2] = g[2]/n;
```

Another necessary data is to calculate the inclination, which indicates how the device screen is rotated relative to the ground. Tilting of the device can be calculated only when the display device is vertically to the ground. This is also supported by the fact that the laying device is in a horizontal level of the screen surface what cannot be displayed.

```
int inclination = (int) Math. Round (Math. To Degrees (Math.acos(g[2])));
```

The last step is a computation of the device tilt according to water level:

```
//computation will run only until 30° from upright – 90°
if (inclination < 120 && inclination > 60)
```

{int rotation = (int) Math. Round (Math. to Degrees(Math.atan2(g[0], g[1])));}

Orientation processing

To calculate the direction of the device according to the north, the development environment has a better ability to process it. To calculate the direction we need not only data from the sensor of geomagnetic field, but also from the accelerometer. The calculation is as follows:

float[] rot Mat = new float[9]; /preparation of field for matrix of rotation devices Sensor Manager. get Rotation Matrix (rotMat, null, magnetic, accel); // Acquisition of matrix, matrix is prepared by parameters, null (or matrix of inclination), a vector of the geomagnetic sensor and accelerometer vector
float[] orient = new float[3]; // preparation of orientation vector of device
Sensor Manager. get Orientation (rot Mat, apr); //acquisition of orientation vector
bearing = apr[0]+(float)(1.5*Math.PI);
// direction is the first component of the vector, the possible transfer of the range 0 to 360 ° (the result is in radians)

Geomagnetic processing

The last important angle used in the application is a calculation of the direction of the target. The following algorithm is based on a modified calculation from [19]. The main idea of the algorithm is that the magnetic sensor gain direction to magnetic north, while the method Location. bearingTo() calculates the direction in degrees from true north. Therefore, it is necessary to calculate the declination indicating the difference in degrees between true north and magnetic north. For this purpose a method in class Geomagnetic Field is prepared, where the first instance need to be created by specifying the current position, altitude and time, then just call of the method get Declination () is needed, which returns the declination in degrees:

Geomagnetic Field geo Field = new GeomagneticField((float)location.getLatitude(),(float)location.getLongitude(), (float)location. get Altitude(),System. current Time Millis());
declination = geo Field.get Declination();

Then we need to calculate the direction to target by the following code, where all angles are specified in degrees:

float bearing = location. Bearing To (locDest); // computation of azimuth to target

location
float heading = Main Activity. heading; // get directions to the north (stored in Main Activity)
Heading +=declination; // modification of the magnetic north on a real north
heading = bearing - heading; // resulting angle is a difference of both angles.

All basic functions of proposed application declared before was achieved. Settings of application is solved by dialogs, which are shown after clicking some area in user interface. Graphics indicators were combined to single composite indicator. This solution maximize usage of space used to show data from sensors [20]. Some major details about GUI of application are described in the next section.

Testing of Augmented Reality Application

For testing purposes of application we used few devices with different hardware configurations and different version of operational system (Table-1).

Table-1. Testing devices (Senzors: T – temperature, P – atmospheric pressure, H – humidity, L – light).

Device	Ver.	Processor	Memory	Resolution	Missing Sensors
SE Xperia Neo	4.2.2	1GHz Arm7	380MB	854x480	T,P,H
SE Xperia Neo	4.0.4	1GHz Arm7	335MB	854x480	L,T,P,H
LG Optimus One	4.0.4	600MHz Arm6	414MB	480x320	L,T,P,H
LG Optimus Black	4.0.4	1GHz Arm7	440MB	800x480	L,T,P,H
HTC Desire Z	2.3.5	800MHz Arm7	368MB	800x480	T,P,H
Samsung G. S3	4.1.2	4x1,4Ghz Arm7	694MB	1280x720	T,H

On all devices application ran correctly, even on the weakest LG Optimus One. For user interface scalability tests was added also some emulator configurations, one with very low resolution 320x240 and second with high tablet resolution 1280x800. Application is usable on all resolutions, except lowest (320x240), where the quality of compound indicator is very low. Functions of application was tested by few users, who did not find some important bugs and application did not crashed.

Some information from sensors could not be tested, because there are almost no devices, which have sensors to provide it. These sensors are humidity and ambient temperature sensors. At the end of testing was also considered memory consumptions and number of threads in Eclipse IDE [17].

DISCUSSION OF RESULTS

Presented project of augmented reality solution presented a various spectrum of sensors embedded to modern mobile devices. At first glance, especially mobile Google Android platform supports a relatively high number of sensors, but other problems appeared that several sensors, which was addressed before, only modifies the software and data from another sensor. Problem with using of built-in sensors is very diverse sensors in support of individual mobile devices. Some mobile platforms such as Apple iOS and Microsoft Windows Phone provide a support from a programming standpoint, for only a few basic sensors. On the other hand, devices on these platforms have specified a mandatory set of sensors, which must be included by developer [21-25].

On the other hand, the Google Android mobile platform offer a wide set of sensors which are all accessible from the development environment. The problem is that Google Android platform will not only specify the required sensors and the developer must always verify their presence, but also the promotion of sensors across the system versions. Some sensors are marked as obsolete and should not be used contrary. A number of sensors can only support the latest version of the system. E.g. humidity sensor and temperature at the time of writing this paper only promote the latest equipment from the top of the pricing spectrum. In the context of a development environment for Google Android platform, we found a problem while using of rapid development which will also come across shortcomings and errors in the documentation as well as inconsistencies in some procedures. E.g. classes within the package classes serving locating device used for geographic coordinates of each pair of double values, while the class to calculate the magnetic declination requires pairs of values of type float. A similar situation prevails even angles, where part of the methods used radians and degrees part.

Big differences are also in reaching out to developers. Here again, smacking best Google Android platform, which is the official development environment available for all major operating systems including emulator.

For publishing applications to business platforms there is a need to register only once for a fee 25USD even to a published application are asked almost no limiting conditions [21-23].

For Apple iOS platform and Microsoft Windows Phone situation is different. The developer must publishing platforms stores in both cases, pay an annual fee of $ 100.

And moreover, which is even more restrictive development environment these platforms only runs on the operating system of the same manufacturer - development for iOS only runs on Mac OS X, which is also available only on computers manufactured directly by Apple.

For Microsoft's offer hardware development broad, but the more restrictions there are for the development, when the full-fledged development including emulator is required for 64-bit Windows 8 Professional and earlier versions are not supported.

Development of model applications after becoming familiar with the development environment play as a practical demonstration of the knowledge we acquired. When working with a number of sensors there are helper methods available to facilitate work with other algorithms. Given the scale of development for Google Android it can be relatively easily to search for different programming sites. On the structure of the applications were from the original design used instead of custom components services that facilitate the implementation of applications and include the base messaging system.

The biggest problem grown during the development process was not realized with sensors, but devising their own dialog for entering coordinates, which are insufficient to standard procedures (Figure-4).

Figure-4. Final GUI of developed augmented reality application – screenshot of dialog after location parameters definition.

The proposed application can be utilized not only as a source of information from the sensors, but more as well as easy to navigate. Application has also been successfully tested on several devices with different hardware and software characteristics and its correct functionality.

Applications could be developed further eg. By adding the possibility of direct shooting from its environment, which would save time for customers switching to the Camera application and could be used as a simple photonotes. The photos would be appropriate to add information and environmental sensors, which would have been enriched with information about the weather at the time of acquisition. On the enriched photos could establish eg. a desktop or web application, which is further processed into clear collections. But now due to the expansion of environmental sensors it would be a very minor issue and only add GPS coordinates can most photo applications as well as the number of applications is able to effectively show.

Using an interesting application could even be used on glasses Google Glass, which for various applications of virtual reality suits [26].

CONCLUSIONS

Firefly algorithm is considering new algorithm in the swarm intelligence family. Despite that, the usage of the firefly algorithm in the various types of problem shows that the anticipation from the researcher to use this algorithm. This algorithm already proves that it is superior compared to the previous introduce swarm intelligence from the research done before. Even though the firefly algorithm has proven to be superior compared to the previous swarm intelligence, some modification can be done to improve the local search a well as global search to ensure the solution obtains is the optimum and not premature solution.

Firefly algorithm also suitable be used for the high dimensional and nonlinear problems. The downside of it is the single that the single metaheuristic is hard to reach an optimal solution within a reasonable time. Thus, by combining the metaheuristic will help to overcome the shortcoming of the single metaheuristic algorithm.

In the future work, the researcher should tackle firefly in more various types of problem such as find optimum route for new build trains rails route which is have multiple constrains such as to preserve the nature maximum as possible and multi tracking for object tracking

REFERENCES

[1] Krejcar O. 2012. Threading Possibilities of Smart Devices Platforms for Future User Adaptive Systems. In 4th Asian Conference on Intelligent Information and Database Systems, ACIIDS 2012, March 19-21, Kaohsiung, Taiwan. LNCS Vol. 7197. pp. 458-467

[2] Behan M. and Krejcar O. 2012. Adaptive Graphical User Interface Solution for Modern User Devices. In 4th Asian Conference on Intelligent Information and Database Systems, ACIIDS 2012, March 19-21, 2012 Kaohsiung, Taiwan. LNCS Vol. 6592. pp. 411-420

[3] Vanus J., Novak T., Koziorek J., Konecny J. and Hrbac R. 2013. The proposal model of energy savings of lighting systems in the smart home care. IFAC Proceedings Vol. 12 (PART 1), pp. 411-415.

[4] Vanus J., Koziorek J. and Hercik R. 2013. Design of a smart building control with view to the senior citizens' needs. IFAC Proceedings Volumes (IFAC-PapersOnline), 12 (PART 1), pp. 422-427. ISSN: 14746670. ISBN: 9783902823533.

[5] Machacek Z., Slaby R., Hercik R. and Koziorek J. 2012. Advanced system for consumption meters with recognition of video camera signal. Elektronika Ir Elektrotechnika. Vol.18, No.10. pp. 57-60. ISSN: 1392-1215.

[6] Behan M. and Krejcar O. 2013. "Modern Smart Device-Based Concept of Sensoric Networks". EURASIP Journal on Wireless Communications and Networking. Vol. 2013, Issue 1, No. 155, DOI 10.1186/1687-1499-2013-155.

[7] Krejcar O., Jirka J. and Janckulik D. 2011. "Use of Mobile Phone as Intelligent Sensor for Sound Input Analysis and Sleep State Detection". Sensors. vol. 11, Iss. 6, pp. 6037-6055, 2011. DOI 10.3390/s110606037.

[8] Benikovsky J., Brida P. and Machaj J. 2012. Proposal of User Adaptive Modular Localization System for Ubiquitous Positioning, Lecture Notes in Artificial Intelligence, Vol. 7197, pp. 391-400

[9] Mlynka M., Brida P. and Machaj J. 2014. Modular Positioning System for Intelligent Transport, 5th International Conference on Computational Collective Intelligence Technologies and Applications, Craiova, Romania, Studies in Computational Intelligence 513, Vol. 513, pp. 115-124.

[10] Penhaker M., Darebnikova M. and Cerny M. 2011. Sensor Network for Measurement and Analysis on Medical Devices Quality Control, Communications in Computer and Information Science, Vol. 171, pp. 182-196.

[11] Daponte P., De Vito, L., Picariello F. *et al.* 2014. State of the art and future developments of the Augmented Reality for measurement applications, MEASUREMENT, Vol. 57, pp. 53-70, Nov.

[12] Gantulga E. and Krejcar O. 2012. Smart Access to Big Data Storage – Android Multi-language Offline Dictionary Application. In 4th International Conference on Computational Collective Intelligence Technologies and Applications 2012, ICCCI 2012, November 28-30, 2012 Ho Chi Minh, Vietnam. Lecture Notes in Computer Science, LNCS Vol. 7653. pp. 375-284.

[13] Kwon O. S., Park C. S., Lim C. R. 2014. A defect management system for reinforced concrete work utilizing BIM, image-matching and augmented reality, Automation in Construction, Vol. 46, pp. 74-81, October.

[14] Rusch M.L., Schall M.C., Lee J.D., Dawson J.D. and Rizzo M. 2014. Augmented reality cues to assist older drivers with gap estimation for left-turns, Accident Analysis and Prevention, vol. 71, pp. 210-221, DOI 10.1016/j.aap.2014.05.020, October.

[15] Dey A. and Sandor C. 2014. Lessons learned: Evaluating visualizations for occluded objects in handheld augmented reality, International Journal of Human-Computer Studies, Vol. 72, Issue 10-11, pp. 704-716, DOI 10.1016/j.ijhcs.2014.04.001, Oct-Nov.

[16] Leu J.S., Su K.W. and Chen C.T. 2014. Ambient mesoscale weather forecasting system featuring mobile augmented reality, MULTIMEDIA TOOLS AND APPLICATIONS, Vol. 72, No. 2, pp. 1585-1609, DOI 10.1007/s11042-013-1462-4, September.

[17] Bilek O. and Krejcar O. 2015. Possibility of Using Embedded Sensors of Smart Devices for Augmented Reality Application. In Lecture Notes in Computer Science Vol. 9073, pp. 488-495.

[18] How to measure the tilt of the phone in XY plane using accelerometer in Android. [cit. 2013-04-19]. from: http://stackoverflow.com/questions/11175599/how-to-measure-the-tilt-of-the-phone-in-xy-plane-using-accelerometer-in-android.

[19] Calculate compass bearing / heading to location in Android. [cit. 2013-04-19]. from: http://stackoverflow.com/questions/4308262/calculate-compass-bearing-heading-to-location-in-android

[20] Longo L. and Kane B. 2011. A Novel Methodology for Evaluating User Interfaces in Health Care, 24th IEEE International Symposium on Computer-Based Medical Systems CBMS 2011, Bristol, England, June pp. 27-30.

[21] Maresova P. 2015. Decision Making Criteria for Cloud Computing Deployment, In Advanced Multimedia and Ubiquitous Engineering, Lecture Notes in Electrical Engineering Vol. 352, 2015, pp. 93-98, Future Information Technology, Hanoi, Vietnam, May 18-19, Proceedings, Springer. ISBN 978-3-662-47486-0.

[22] Maresova P. and Klimova B. 2015. Investment evaluation of cloud computing in the European business sector, Applied Economics, Vol. 47, No. 36, pp. 3907-3920, 2015

[23] Maresova P. and Halek V. 2014. Deployment of Cloud Computing in Small and Medium Sized Enterprises in the Czech Republic, E & M Ekonomie a Management, Vol. 17, No.4, pp. 159-174.

[24] Horalek J., Matyska J. and Sobeslav V. 2013. Communication protocols in substation automation and IEC 61850 based proposal, CINTI 2013 - 14th IEEE International Symposium on Computational Intelligence and Informatics, Proceedings, art. no. 6705214, pp. 321-326.

[25] Cimler R., Matyska J., Balík L., Horalek J. and Sobeslav V. 2015. Security issues of mobile application using cloud computing. Advances in Intelligent Systems and Computing, Vol. 334, pp. 347-357.

[26] Bodnarova A., Sobeslav V., Horalek J., Matyska J. and Hatas M. 2013. Virtual laboratory. In: ICETA 2013: proceedings. Piscataway: IEEE, s. 53-58. ISBN 978-1-4799-2161-4.