Ultrasonic indoor positioning for smart environments: a mobile application

E. Díaz , M. C. Pérez, D. Gualda, J. M. Villadangos, J. Ureña, J. J. García
Electronic Department, University of Alcala, E-28805, Alcala de Henares (Madrid), Spain
edel.diaz@depeca.uah.es; mcarmen.perezr@uah.es

***Abstract*—In this paper we present a mobile application and solution for accurate smart indoor positioning. Smart society applications do normally require user location, specifically, in indoor environments high accuracy can enrich augmented or virtual reality, gaming, in-building guidance or support for ambient assisted living. We use encoded ultrasonic signals and TDMA protocol to obtain fine-grained distance measurements. Signals are emitted from a set of low cost ultrasonic local positioning systems, operating around 41kHz. An acquisition module, based on a MEMs microphone and a microcontroller, digitizes at 100kHz the incoming signals and send them over an USB protocol to the mobile device for their processing. We have implemented an Android Application that computes the Time Difference of Arrival (TDOA) to estimate the current position and display it in the mobile screen. Several users can compute their positions autonomously and user privacy is protected. The application can be configured for different encoding techniques and modulation schemes according to the environment requirements. Absolute error less than 5 cm is achieved in a 5x6m complex environment in 85% of the cases for an average position refresh period of 200ms.**

Keywords—indoor positoning, Android application, LPS, smart indoor environments

I. Introduction

People spend 80%-90% of their time in indoor environments [1]. On the other hand, the majority of people own a smartphone or tablet in advanced economies, [2] report 72% of adults in United States, percentage that increases to 92% if only people aged from 18 to 29 is considered. Similar numbers can be obtained for several European countries. This has triggered the research in indoor positioning so as to overcome the GPS limitations inside buildings and offer an extended variety of Location Based Services (LBS) to users. Some interesting applications include medical professionals and patients tracking inside the complex environment of hospitals; tourist navigation inside museums to offer adaptive personal tracking guide depending on the user behavior; gaming, augmented and virtual reality, etc. See [3] for further scenarios where indoor positioning is required to support LBS.

A wide variety of technologies have been used for the design of indoor positioning systems, such as Radio-Frequency identification (RFID), Ultra-Wideband (UWB), video cameras, Bluetooth sensor networks, ultrasounds and audible sound. Each of them offer different accuracies (a good survey can be found in [4] and [5]). Among them, ultrasonic signals can be interesting for fine grained positioning as they provide centimeter accuracies and do not require expensive infrastructure.

In [6] an Ultrasonic Local Positioning System (U-LPS) for smartphone positioning is proposed. It consists of modulated ultrasonic chirp signals sweeping from 19 to 23 kHz that are emitted by a set of transmitters broadcasted over the environment. Then, any number of mobile devices able to record sound inside the 19-24 kHz range is intended to compute its own position by time difference of arrival (TDOA) pseudo-random technique. Authors have tested different mobile devices finding some limitations above 22kHz. A similar idea is proposed in [7], that uses BPSK modulated Kasami codes emitted at 20kHz. Codes are captured by an iOS device at 48kHz sampling rate for their processing. The limited sampling frequency and narrow bandwidth for transmitting the encoded signals decrease the signals quality and audible artifacts can appear.

To overcome that difficulties, in [8] we presented an indoor positioning system based on a U-LPS that simultaneously emits encoded signals at 41.67kHz. An external acquisition module based on a MEMs microphone digitizes the incoming signals at 100kHz and send them over a USB connection to the mobile device. Then, an Android application was programmed for all the data processing involved in the position estimation, namely: identification of the emitted codes by matched filtering, correlation peak detection, TDOA computing and Gauss Newton algorithm to solve the user's location.

This paper presents an improved version of the work in [8]. On the one hand, we have changed the control unit of the U-LPS from a FPGA board [8] to a LPC1768 microcontroller [9] so as to reduce the control unit board dimensions (from 20x26 cm to 11x11 cm) and cost. Thus, the emission pattern and former TDOA computation has been adapted to the demands of the new hardware design. On the other hand, we have also improved the map performance, replacing the previous XML map for Google Maps. Further features have been added and the processing has been optimized to significantly reduce the computation time. Execution time is critical in the design, since all computations and results have to be done and displayed in real time, considering an average human walking speed of 1.38m/s.

The remainder of the paper is structured as follows: Section II describes the general system; Section III details the Android application improvements; Section IV presents the obtained results in real tests in terms of location accuracy and execution time; and finally, the main contribution of this work together with the description of possible further improvements are summarized in Section V.

978-1-5386-0810-4/17/$31.00 ©2017 IEEE

II. System Overview

A. U-LPS configuration

Fig. 1 depicts a diagram of the proposed system. The U-LPS consists of five ultrasonic emitters, distributed around a 70.7x70.7cm square structure. The coverage area is of 30m^2 when it is installed at the ceiling at a height of 3m. Five different 255-bit Kasami sequences have been used to accurate identify every emitter of the U-LPS [10]. The codes have been BPSK modulated with two periods of a sinusoidal carrier at 41.67kHz, in order to adapt their spectral features to the frequency response of the ultrasonic emitters (Prowave 328St160 [11]). The modulated codes have been previously generated offline and sampled at 500kHz, so 12 samples per carrier period are considered. If $\{K=k_m[l] \in \{-1,1\}; 0 \leq m \leq M-1; 0 \leq l \leq L-1\}$ is a family of Kasami codes of length $L=255$, in which the $M=5$ codes with better correlation properties are selected for encoding the emitters of the U-LPS, and $s[n]$ is the modulation symbol with $N_c=2$ carrier cycles and oversampling factor equal to $O_f = \frac{f_s}{f_e} = \frac{500kHz}{41.67kHz} = 12$, the modulated pattern to be emitted can be represented as (1):

$$sk_m[n] = \sum_{i=0}^{L-1} k_m[i] \cdot s[n - i \cdot N_c \cdot O_f] \qquad (1)$$

The modulated codes are generated off-line in a Personal Computer and stored in memory of the U-LPS microcontroller that manages all the emission process [9]. All parameters regarding the code design (type of code, length and oversampling factor) can be easily configured according to the demands of the final application.

Since only one Digital to Analog Converter (DAC) is available in the microcontroller, a time division multiple access (TDMA) technique has been implemented, which also reduces multipath and multiple access interferences (MAI). A slot time of 13.94ms has been devoted for the transmission of every code. Thus, the time interval between two consecutive transmissions of the same code is of 69.7ms (see Fig. 2).

B. Receiver design

Commercially available mobile devices, such as smartphones, have some hardware restrictions regarding the maximum sampling rate (44.1kHz) and maximum supported frequencies (some of them can have 20kHz low-pass filters) what limit the performance with ultrasonic signals [12]. To overcome that, signals were not directly captured by the mobile device, but with an attached acquisition module, as Fig. 3 shows. The acquisition module is based on low cost components, and includes a MEMS SPU0414HR5H-SB [13] microphone with an adequate bandwidth at 41.67kHz. Signals captured by the microphone are high-pass filtered and the gain is adjusted depending on the application requirements and signal level. A STM32F103 microcontroller [14] digitizes the incoming ultrasonic signals at a sampling rate of 100kHz. Note that a decimation by 5 has been included to reduce the amount of data to be processed.

A buffer with 0.08192s of the ultrasonic signal (8192 samples) is stored and send over a USB link to the portable device. In a previous version of the receiver acquisition module [8] we used a Bluetooth communication protocol to avoid a wired connection, nevertheless we needed an external battery to supply the receiver. Hence, to simplify the hardware and to obtain a faster and more reliable connection we finally have chosen an USB connection.

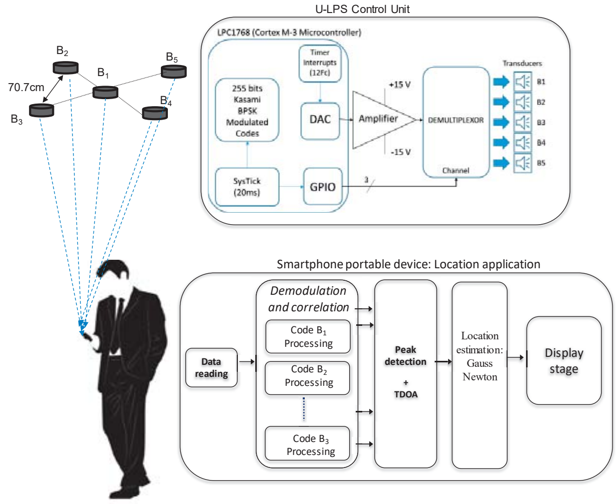

Fig. 1.- General view of the proposed system.

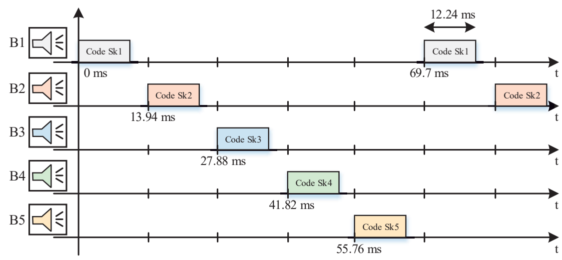

Fig. 2.- Beacon emission pattern.

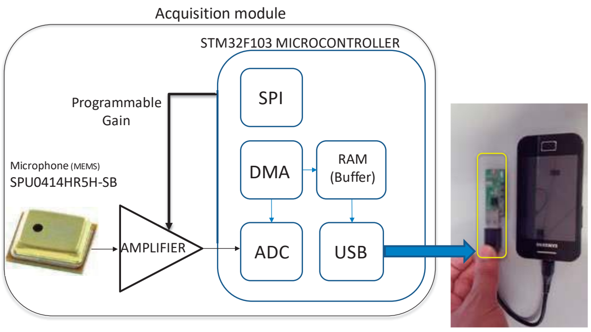

Fig. 3.- Receiver acquisition module.

The receiver buffer size assures that one complete emission from each of the beacons is received. As an example, Fig. 4 represents the reception corresponding to a beacon B_i stored in the acquisition buffer in four different time intervals. Note that in all cases one complete reception of the beacon is stored for its later correlation. On the other hand, the value of 8192 samples is limited by the FFT function implemented in Android, which has to be power of two.

In the portable device, an Android application provides mathematical functions for all the data processing involved in the position estimation and screen visualization.

III. Software Design

Hereafter, it is presented a new version of the software application in [8], named Locate-US, to answer to the hardware specifications detailed in Section II-A and to reduce the execution time of the signal processing and display stage.

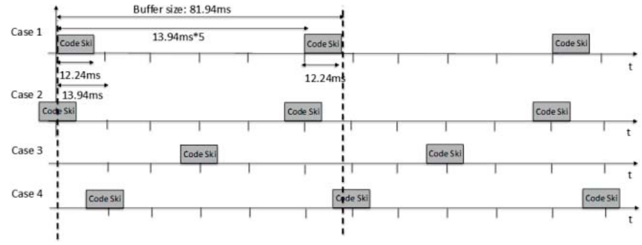

Fig. 4. – Example of cases in which the buffer acquires the modulated code corresponding to a beacon Bi in different time instants.

Specifically, this version is able to work with TDMA protocol and allows the screen visualization of the correlation peaks and the execution time of the low level tasks involved in the position estimation. Other advantage is the improvement in map performance, the XML map has been substituted by Google Maps what significantly increases the speed in the representation of the estimated position in the building map.

Locate-US application can be divided into three different stages (see Fig. 5). Some of the processes (threads) run in parallel, avoiding the collapse of the user interface and reducing the execution time of the application. Also, a Fragment has been added to the activity in order to visualize the position in the map.

In the first stage, the global variables and templates to be used during the application are loaded. This information includes the number and position of beacons, and the complex conjugate of the FFT (Fast Fourier Transform) of the signal pattern that encodes every beacon, which is previously generated offline. Also, the stage includes the data reception over an USB connection. When the input buffer of the mobile device is full, an Android primitive function is called and the 8192-sample-long input signal is read.

In a second stage, the correlation is obtained as the inverse FFT of the product of the FFT of the received signal by the complex conjugate of the FFT code pattern, which has been stored in memory during the setup stage. Then, the main correlation peaks are detected and the TDOA are obtained to compute the location by a Gauss-Newton algorithm. Only when the position is ready to be depicted, the Fragment actualizes the position map. The Fragment behaves like a process in parallel avoiding lags in the application.

Finally, the trajectory is shown in the mobile device screen. Apart from the calculation and representation of the device location, Locate-US has a setting menu that allows to modify several variables according to the requirements of the application or the user preferences: type of code and code length, number of carrier cycles, oversampling, slot time and

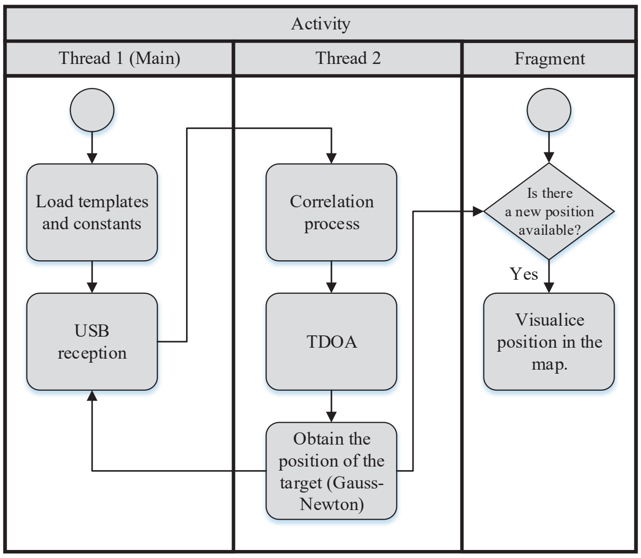

Fig. 5. – LocateUS flowchart.

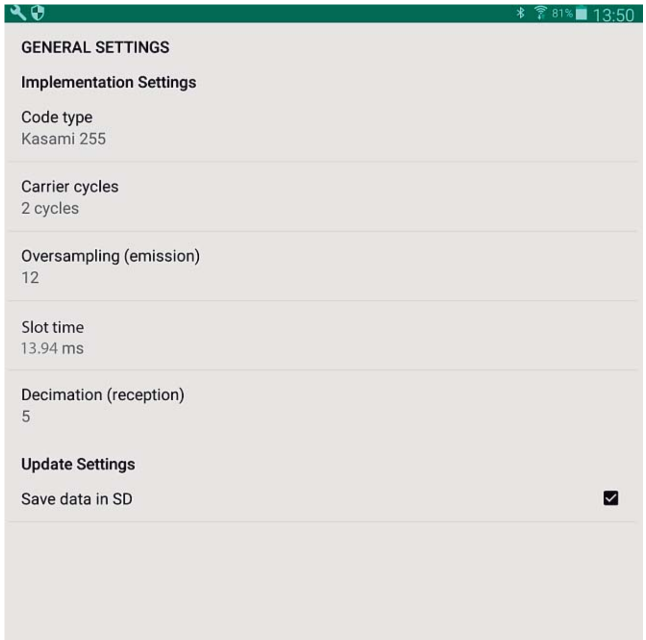

Fig. 6. – Settings menu.

decimation in reception (see Fig. 6). Also it allows to save the input signal in a SD card.

A. Low level data processing and position estimation

As explained before, the application identifies the emitted codes performing a cross-correlation (by means of the FFT) of the acquired signal with the emission patterns. A correlation peak indicates the time instant of arrival of each of the emissions. Since every beacon is encoded with a different code, the origin of the correlation peak can be clearly identified. To correctly compute the TDOA between beacons, the TDMA protocol used in the emission stage has to be considered. First, the application looks for the maximum values of the correlation functions, and identify the number of sample that corresponds to the first received code, which acts as a reference. Then, the number of samples associated to the remaining correlation peaks from the other beacons are subtracted from that of the reference beacon. As well as this, the fixed time interval between the emission of the reference beacon and the other beacons has to be subtracted, so as to make the TDOA only depending on the relative position of the user and the beacons. See Fig. 7 for an example, where code 5 is taken as a reference (number of sample: 466) and the time difference with regard beacon 2 (number of sample: 3273) is computed as: - 466 + 3273 - 1394 * 2 = 19 samples. Note that the number 1394 corresponds to the 13.94 ms interval between emissions of adjacent codes.

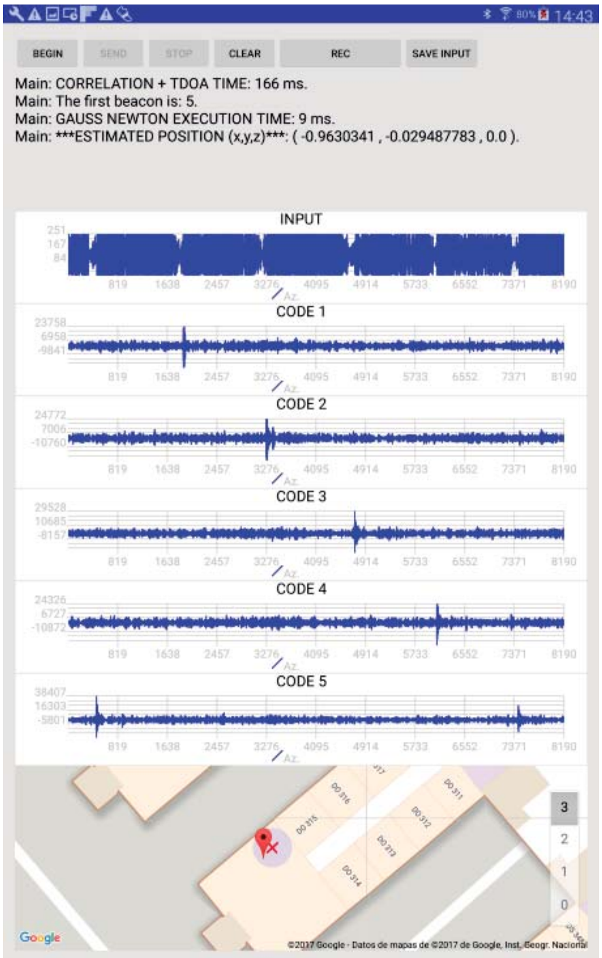

Fig. 7 – Screen photo of the Locate-US application.

In the next step, the difference in samples is translated into time by multiplying it with the inverse of the sampling frequency (100 kHz). Finally, from this time, the distance in meters between the mobile device and each beacon is obtained by means of a Gauss-Newton trilateration algorithm. The Gauss-Newton algorithm considers the TDOA and the last position estimated and it obtains the position within a maximum of 100 iterations, however, in most cases, the number of iterations necessary to obtain the position is less than 10. The limit of iterations is fixed by the maximum time of execution allowed in our design. The application provides the estimated positions and the execution time of all processes involved (see top of Fig. 7.)

B. *User Application*

The designed user application has 3 blocks to be depicted in the screen. The first block is the obtained result, which shows the estimated positions results and execution time. The second shows the plots of the correlations. Finally, the location in the building map is displayed by means of a red map pointer icon (see Fig. 7). In the map, the red cross indicates the position of the U-LPS and the gray circle its coverage area.

Due to the high resource consumption of the correlation representation in the screen of the mobile device, it is possible to deactivate the representation in the setting menu.

IV. Results

The application is executed in a tablet Samsung Galaxy S Tab, that has a microprocessor QuadCore 1.9 GHz and 3 GB of RAM memory. The acquisition module to capture the ultrasonic signals has been detailed in Section II-B, if more information is required see [8] and [15].

The experiment has been done in a laboratory of the Electronics Department of the University of Alcala (see Fig. 8 for the experimental set-up). It is a complex environment, with furniture and other scattering objects that can cause problems because of multipath effect. The U-LPS is placed at the ceiling at a height of 3 m. 5 test points (P1 to P5) has been considered and 40 measurements were taken from each of these positions to evaluate the performance of the system.

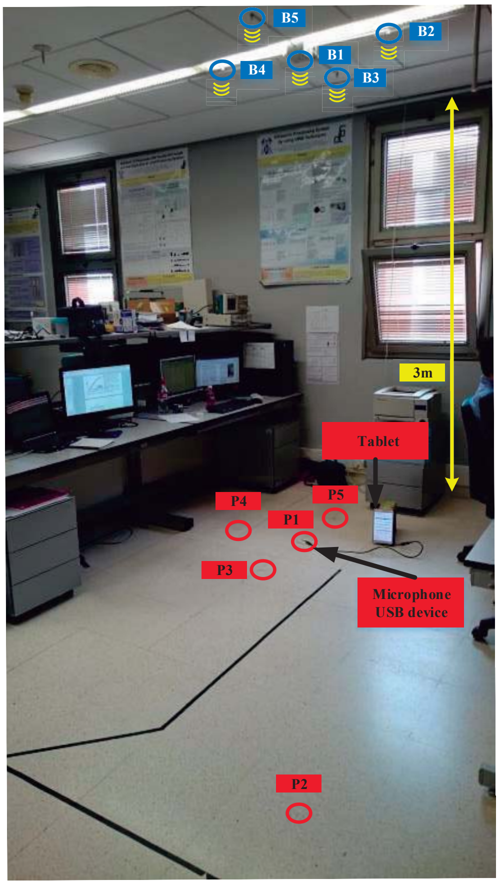

Fig. 8. – Distribution of the static test in the experimental set-up.

The Cumulative Distribution Function (CDF) of the error considering the test over all the points is shown in Fig. 9. It can be observed that in 85% of the cases the error is less than 5 cm in X coordinate, and in 90% of the cases for Y coordinate. On the other hand, Fig. 10 shows the estimated positions for each test position, as well as the distribution of the 95% of the errors for each point represented by an ellipse error. P3 has the lowest error because it is less affected by multipath effect and it is placed just under the beacons. On the contrary, P5 point is the most affected by the multipath effect since it is close to a wall and presents a greater error than P1, P3 or P4. The P2 position is the one that achieves the worst results due to its location, far away from the beacons. The CDF results of P2 point indicates that the error is less than 15 cm in 80% of the cases for the X coordinate, and in 95% of the cases for the Y coordinate. In addition, a study of the execution time of each process has been carried out. The measured execution times are: Correlation time, TDOA computing time and time to perform the Gauss-Newton algorithm. The correlation is the most time

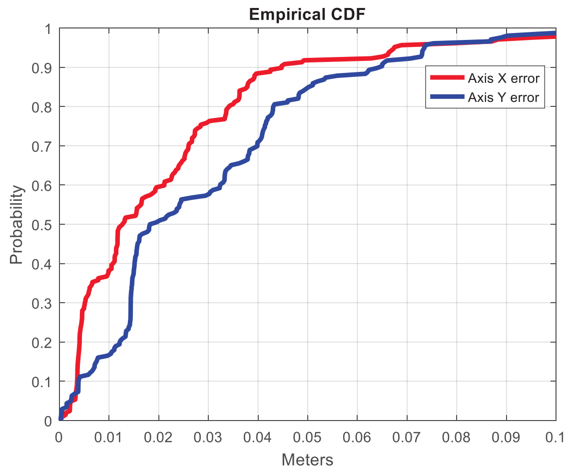

Fig. 9. – CDF of the error for the static test positions of Fig. 8.

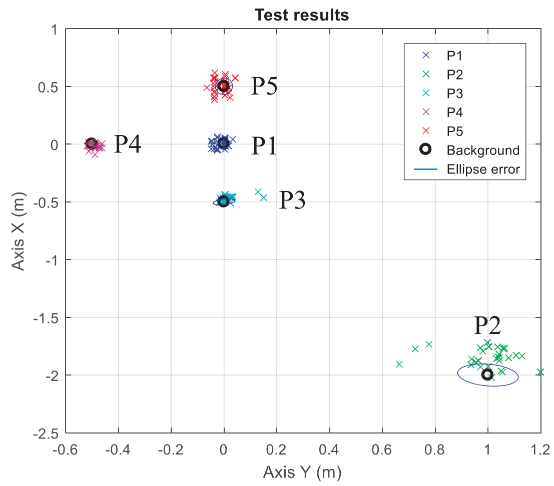

Fig. 10. – Test results and distribution of the 95% of the errors for each point.

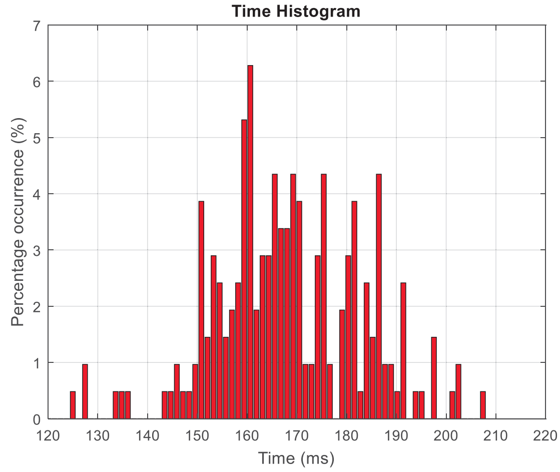

Fig. 11. – Execution time histogram for position estimation: correlation, TDOA and Gauss Newton algorithm.

consuming process (98% total time), meanwhile the TDOA computing requires less than 1 millisecond. Fig. 11 depicts the histogram of the total execution time during the tests. The average time obtained is of 167ms and the maximum time required is 207ms.

Table I compares the presented work and the previous version in [8] with regard to the execution time. As can be seen, in average, the execution time has been improved 49% and the standard deviation is 79% lower.

TABLE I: COMPARISON OF EXECUTION TIME FOR A POSITION UNDER THE BEACONS (P1).

	New version	*Previous version [8]*
Execution time (ms)	167	327
Standard deviation of the execution time (ms)	17	81

V. CONCLUSIONS

This paper presents a U-LPS for accurate positioning of Android mobile devices, such as smartphones or tablets. It is based on a TDMA protocol to transmit encoded ultrasonic signals that clearly identify every beacon of the U-LPS. The receiver includes a MEMs microphone and a microcontroller that digitizes the incoming signals and send them over a USB protocol to the mobile device. An Android application that allows the position estimation in real time has been presented.

The application can be configured for different scenarios adjusting the type of code or the code length. It also allows the visual inspection of correlation functions in the device screen, as well as the analysis of the execution time. The estimated position of the mobile device in the building map is also depicted. A study of the reliability of the system in certain positions of a research lab is included. Considering 40 measurements in every one of the five test positions considered, errors bellow 5cm are obtained in 85% cases, with overall execution time of the algorithms in the intelligent device below 0.2s. Thus, the application can run in real time.

Further works should improve the execution time of the correlations and include pedestrian dead reckoning system (PDR) to merge the U-LPS information with the relative location in those cases with low U-LPS coverage.

ACKNOWLEDGMENT

This work has been supported by the University of Alcala (LOCATE-US project, ref. CCG2016/EXP-078) and the Spanish Ministry of Economy and Competitiveness (TARSIUS, TIN2015-71564-c4-1-R); and thanks to the National Youth Guarantee System Administrative, Unit of the European Social Fund (Order ECC / 1402/2013).

REFERENCES

[1] L. K. Wang, M. H.S. Wang, "Handbook of Environmental Engineering", Springer, 2016.

[2] J. Pushter, "Smarthpnone Ownership and Internet Usage Continues to Climb in Emerging Economies", Pew Research Center, February 2016.

[3] Y. Gu, A. Lo, I. Niemegeers, "A Survey of Indoor Positioning Systems for Wireless Personal Networks", IEEE Communications Surveys & Tutorials, vol. 11, no. 1, 2009.

[4] V. Filonenko, C. Cullen, J. D. Carswell, "Indoor positioningt for smartphones using asynchronous ultrasound trilateration,"ISPRS International Journal of Geo-Information, vol. 2, pp. 598-620, 2013.

[5] D. Lymberopoulos, J. Liu, X. Yang, "A Realistic Evaluation and Comparison of Indoor Location Technologies: Experiences and Lessons Learned, " Proc. Of the 14th International Conference on Information Processing in Sensor Networks, pp. 178-189, 2015.

[6] P. Lazik. A. Rowe, "Indoor Pseudo-ranging of Mobile Devices using Ultrasonic Chrips", Proceedings of the 10th ACM Conference on Embedded Network Sensor Systems, pp. 99-112, 2012.

[7] T. Aguilera, J. A. Paredes, F. J. Álvarez, J. I. Suárez, A. Hernández, "Acoustic Local Positioning System Using an iOS Device", Proceedings of the 2013 International Conference on Indoor Positioning and Indoor Navigation, pp. 1-8, October, 2013.

[8] M. C. Pérez, D. Gualda, J. M. Villadangos, J. Ureña, P. Pajuelo, E. Díaz, E. García, "Android application for indoor positioning of mobile devices using ultrasonic signals", Proceedings of the 2016 International Conference on Indoor Positioning and Indoor Navigation, October, 2016.

[9] LPC1768 Product Data Sheet, http://www.nxp.com/documents/data_sheet/LPC1769_68_67_66_65_64_63.pdf, last access September 2016.

[10] T. Kasami, "Weight distribution formula for some class of cyclic codes", Combinational Mathematics and its Appl., Chapel Hill, N.C.: University of North Carolina Press, 1969.

[11] Pro-Wave Electronics Corporation, Air Ultrasonic Ceramis Transducers 328ST/R160, Product Specification, 2014.

[12] S. Lopes, J. M. N. Viera, J. Reis, D. Alburquerque, N. B. Carvalho, "Accurate Smartphone indoor positioning using a WSN infraestructure and non-invasive audio for TDoA estimation", Pervasive and Mobile Computing, vol. 20, pp. 29-46, July, 2015.

[13] Knowles Aoustics Application Note, SiSonic Design Guide, http://media.digikey.com/pdf/Data%20Sheets/Knowles%20Acoustics%20PDFs/SiSonic_Design_Guide.pdf, last access September 2016.

[14] STM32F103 datasheet, ST life.augmented, www.st.com/resource/en/datassheet/stm32f103tb.pdf, last accessed february 2016.

[15] A. Lindo, M. C. Pérez, J. Ureña, D. Gualda, E. García, J. M. Villadangos, "Ultrasonic signal acquisition module for smartphone indoor positioning," in Proc. of 2014 IEEE Emerging Technology and Factory Automation, ETFA'14, Barcelona, Spain, September 2014.