Contextualized Hand Gesture Recognition with Smartphones

Enrique Garcia-Ceja, Ramon Brena, and Carlos E. Galván-Tejada

Tecnológico de Monterrey, Campus Monterrey, Av. Eugenio Garza Sada 2501 Sur, Monterrey, N.L., México
{A00927248,ramon.brena}@itesm.mx, ericgalvan@uaz.edu.mx

Abstract. Most of the previous works in hand gesture recognition focus in increasing the accuracy and robustness of the systems, however little has been done to understand the context in which the gestures are performed, i.e, the same gesture could mean different things depending on the context and situation. Understanding the context may help to build more user-friendly and interactive systems. In this work, we used location information in order to contextualize the gestures. The system constantly identifies the location of the user so when he/she performs a gesture the system can perform an action based on this information.

Keywords: hand gesture recognition, context-aware, accelerometer, Wifi, smartphone, DTW.

1 Introduction

In recent years, people have been devising new ways to communicate and interact with machines, e.g., voice commands, touch screens, gestures, etc. Coupled with this, the use of devices with sensors like accelerometers, gyroscopes, magnetometers, among others, has increased rapidly. Examples of this devices are smartphones, smart watches [1], tablet pc's, fitness monitoring bracelets [2], etc. In this work we focus on wearable sensors for gesture detection. Advantages of using wearable sensors over video cameras for gesture recognition are that they do not require a fixed infrastructure, their range is not limited to a particular area and they raise less privacy concerns. For the rest of this work, a *hand gesture* refers to the movement of the hand and the arm to form certain patterns. The difference with pure hand signals is that in this form of communication, the position of the fingers is relevant to identify the overall hand shape.

Most of the previous works in hand gesture recognition focus in increasing the accuracy and robustness of the systems, however little has been done to understand the context in which the gestures are performed, i.e, the same gesture could mean different things depending on the context and situation. Understanding the context may help to build more user-friendly and interactive systems. For example, a set of gestures $\{g_1 ... g_n\}$ may be used to turn on the lights in different locations inside a house. In this case the user would have to memorize a gesture for each light he would like to turn on. It may be more practical to have

J.F. Martínez-Trinidad et al. (Eds.): MCPR 2014, LNCS 8495, pp. 122–131, 2014.

a single gesture to perform that action and let the system choose which light to turn on depending on the context of the person.

In this work, we used location information in order to contextualize the gestures. The system constantly identifies the location of the user so when he/she performs a gesture the system could perform an action based on this information. The user's location is identified using the on-range Access Points and their signal strength, the gestures are recognized using Dynamic Time Warping [3]. A smartphone was used to implement a prototype, and unlike other systems that send the information to a server to perform the computations, here, all the processing is made inside the smartphone.

The rest of this document is organized as follows: Section 2 presents the related work. In Sect. 3 we describe the sensing platform we used to collect the data and the data collection process for gestures and locations. Section 4 explains the architecture and design of the system. Section 5 describes the gesture recognition process. Section 6 presents the method we used to identify the user's location. Section 7 describes our experiments and results and finally in Sect. 8 we draw conclusions and propose the future work.

2 Related Work

In recent years, there have been several works in accelerometer based gesture recognition. Specifically, in [4] they used Bayesian classification and Dynamic Time Warping (DTW) to classify 4 gestures (circle, figure eight, square, star) achieving accuracies of 97% and 95% respectively. In [5] they proposed a Frame-based Descriptor and multi-class Support Vector Machine approach with a recognition rate of 95.21% for 12 gestures in the user-dependent case and 89.29% in the user-independent case. They used a Wiimote[1] and the processing was made in a laptop. Akl and Valaee [6] used dynamic time warping, affinity propagation and compressive sensing achieving an accuracy of 99.79% for user-dependent recognition using 18 gestures. For the user-independent recognition, they achieved an accuracy of 96.89% for 8 gestures, however they asked the participants to keep the remote straight. We believe that with these recognition rates it is already possible to implement non critical real world systems for home automation, entertainment, appliance control, etc. There is also a recent work called WiSee [7] in which they used wireless signals such as Wifi to perform the recognition so the user does not need to carry any type of device. Kühnel et al. [8] did a very complete work about gesture recognition for controlling home appliances. They conduct a survey and found that the majority of the respondents liked the idea of controlling appliances with gestures. Also, the majority answered that they would prefer a predefined set of gestures that they could adjust instead of designing their own. Their implementation was made on a smartphone that includes a user interface to select the devices. They also conducted experiments to find a good gesture vocabulary and studied their memorability.

[1] Wii `http://www.nintendo.com/wii`

This work differs from the above mentioned in the sense that we are also taking context into account in order to perform an appropiate action based on the detected gesture.

3 Sensing Platform

An LG Optimus Me smartphone was used to collect the accelerometer and Wifi data. This smartphone has a STMicroelectronics triaxial accelerometer. It returns the acceleration value for each of the axes (x,y,z). Its maximum range is $\pm 19.60 m/s^2$. The x-axis runs parallel to the width of the smartphone, the y-axis parallel to the height of the phone and the z-axis perpendicular to its face (see Fig. 1).

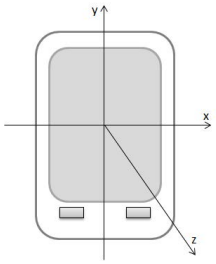

Fig. 1. Acceleration axes

3.1 Data Collection

Gestures. First, we collected the data for 10 different gestures (triangle, square, circle, a, b, c, 1, 2, 3, 4) from 10 persons. We did not instruct the participants to hold the cellphone in any particular manner. These gestures were selected because they encompass three groups of interest: shapes, letters and digits and this is a first attempt to the recognition of the entire set of letters from the english alphabet, the set of decimal digits and a larger set of shapes in a continuous manner(i.e, being able to recognize a sequence of letters and digits to form words and numbers which will be left for future work). Each person performed 5 repetitions of each of the gestures. The sampling rate was set at 50 Hz. To record a gesture the user presses the phone screen with the thumb, performs the gesture and stops pressing the screen. At the end of the process each gesture is represented by the accelerations in each of the three axes (Figure 2).

Locations. The data collection process consisted of generating several instances for every room we want to recognize. To generate one instance we scan the room and record the BSSID (Basic Service Set Identifier) and signal strengths of the detected Access Points, then we perform two more scans with a delay of 500 ms between the scans. The reason behind doing several scans is because in [9] they observed that sometimes one or more Access Points may not be detected

Fig. 2. Performing a triangle gesture

because limited sensitivity of the hardware and/or long beacon interval of some Access Points. Each instance has a List L in which each element is a pair $\langle$*bssid, signal strength*$\rangle$ where *signal strength* is the mean signal strength of the 3 scans for that specific BSSID. For each location, we collected 3-4 minutes of data.

4 System Design

The overall system architecture is illustrated in Fig. 3 and consists of four main components: the cellphone application, a Ninja Block[2], Ninja Blocks REST API and the actuators. The cellphone application has three modules: *Gestures, Location* and *Inference*. The functioning of the first two modules is explained in Sections 5 and 6, respectively. The *Inference* module is made up of simple conditional rules. For future work we intend to replace it with some type of probabilistic inference method. Once the user performs a hand gesture, the *Inference* module uses this information along with the current location to select an action, e.g, turn on the lights, activate an alarm, etc. The action is sent in the form of a command to the Ninja Blocks REST API which in turn sends it to the Ninja Block and finally it activates the corresponding actuators. A Ninja Block is a hardware box that makes it easy to build applications that talk to hardware. A video of the prototype is available at `http://youtu.be/47-35YmimN4`.

5 The Gesture Recognition Process

Every time a gesture is performed the phone may be rotated in a slightly different way. To account for variations of this type we compute the magnitude of the three accelerations in order to work in just one dimension.

$$Magnitude(t) = \sqrt{a_x(t)^2 + a_y(t)^2 + a_z(t)^2}, \tag{1}$$

where $a_x(t)$, $a_y(t)$ and $a_z(t)$ are the accelerations at time t.

[2] Ninja Blocks website `http://ninjablocks.com/`

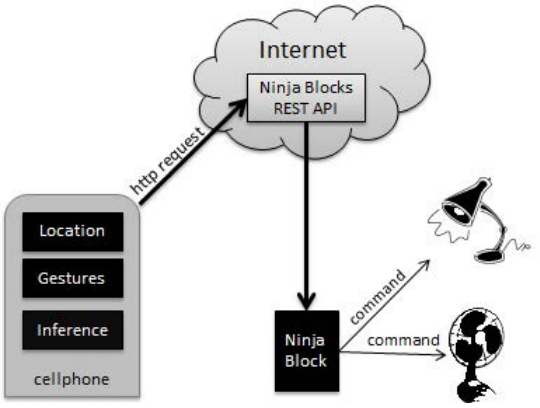

Fig. 3. Overall system architecture

We will call the *query instance* Q the gesture we want to recognize and a *reference instance* R a gesture from the training set of which we know its true class. We used Dynamic Time Warping (DTW)[3] to compute the dissimilarity between the query instance Q and every reference instance R from the training set and return the statistical mode of the k least dissimilar reference instances' class as the predicted gesture. DTW is a method that finds an optimal match between two given time-dependent sequences. It also computes the dissimilarity between those sequences, i.e, if the sequences are the same the dissimilarity will be 0. Let $X = (x_1, x_2, ..., x_{T_x})$ and $Y = (y_1, y_2, ..., y_{T_y})$ be two sequences where x_i and y_i are vectors; $d(i_x, i_y)$ the dissimilarity between vectors x_i and y_i; ϕ_x and ϕ_y are the warping functions that relate i_x and i_y to a common axis k:

$$i_x = \phi_x(k), k = 1, 2, ..., T \tag{2}$$

$$i_y = \phi_y(k), k = 1, 2, ..., T \tag{3}$$

Thus the global dissimilarity is:

$$d_\phi(X, Y) = \sum_{k=1}^{T} d(\phi_x(k), \phi_y(k)) \tag{4}$$

and the problem consists of finding the warping functions that minimize Eq.(4), that is:

$$\min_{\phi} d_\phi(X, Y) \tag{5}$$

which can be solved efficiently using dynamic programming.

Figure 4 shows the alignment produced by DTW for two triangle gestures with a resulting absolute dissimilarity of 87.30. For the prototype, we used Stan's

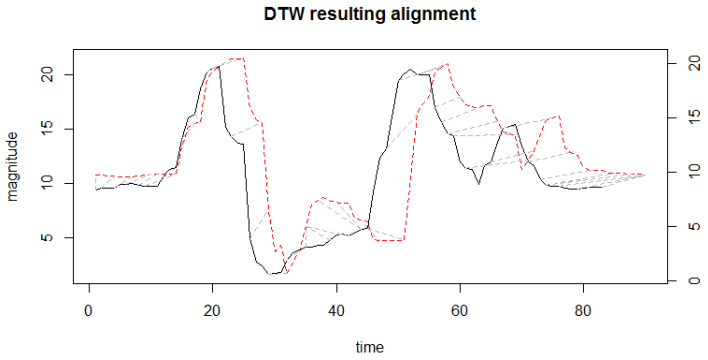

Fig. 4. Resulting DTW alignment. The query instance is represented by the solid line and the reference instance by the dotted line.

approximate DTW algorithm implementation [10] which has linear time and space complexity unlike the exact method which has quadratic complexity.

6 The Location Identification Process

The number of Wifi Access Points around the world has increased significantly in the last years. They are installed in many places such as restaurants, hotels, schools, parks, airports, etc. Since every Access Point has a unique identifier namely, the BSSID (Basic Service Set Identifier), it is possible to use this information along with the signal strength for localization and tracking purposes [11, 12, 13]. In this work we are not focused in computing the absolute or relative location (coordinates on a plane) of the user. Instead, we want to know in which of the n rooms that compose the house or apartment the user is in so this can be seen as a classification problem. Since we do not need the exact coordinates, we can take advantage of the existing Wifi Access Points without any further configuration, i.e., we do not need to know the location of each Access Point.

Once a query instance is generated (as described in Section 3.1), the classification is done using K-Nearest Neighbors with $k = 3$ and distance function $d(q, r) = j(q, r) + s(q, r)$ where q is the query and r is a reference instance from the training set.

$$j(q, r) = 1 - \frac{|L(q) \cap L(r)|}{|L(q) \cup L(r)|} \tag{6}$$

where the function L returns the list of access points of the specified instance. Eq.(6) is known as the Jaccard distance [14].

$$s(q, r) = 1 - (1/1 + \alpha) \tag{7}$$

where $\alpha = abs(SD(q,r) - SD(r,q))$ and $SD(p1,p2)$ is a function that returns the standard deviation of the signal strength of all access points of *p1* that are also in *p2*.

7 Experiments and Results

In this section, we describe the hand gesture recognition experiments and results and then, the location experiments and results.

7.1 Hand Gestures Experiments and Results

Two types of tests which are common for validating the accuracy of gesture recognition systems were performed: user-dependent case and user-independent case. The former consists of evaluating the recognition accuracy by training and testing the system with data from the same user. The latter consists of testing the system with data from the user to be evaluated and training the system with data from all other users. The recognition accuracy for each of the persons was evaluated using leave-one-out cross validation [15] for the user-dependent case (Table 1). For testing the user-independent case, leave-one-person-out cross validation [5] was used. The average accuracy for the user-dependent case was 93.8% and for the user-independent case it was 78.4%. Figure 5 shows the confusion matrix for both, the user-dependent and user-independent case. In matrix (a) we can see that shape 'a' was confused 4 times with a circle, 2 times with shape 'b', 1 with 'triangle', 1 with '3' and 1 with '2'.

Table 1. Hand gesture recognition results

Person	user-dependent	user-independent
1	88%	80%
2	90%	82%
3	94%	84%
4	96%	90%
5	94%	72%
6	98%	72%
7	96%	50%
8	92%	86%
9	92%	90%
10	98%	78%
average	93.8%	78.4%

7.2 Location Experiments and Results

The location recognition was tested in 4 different scenarios; two apartments and two houses. In each scenario different locations were chosen to be recognized. For

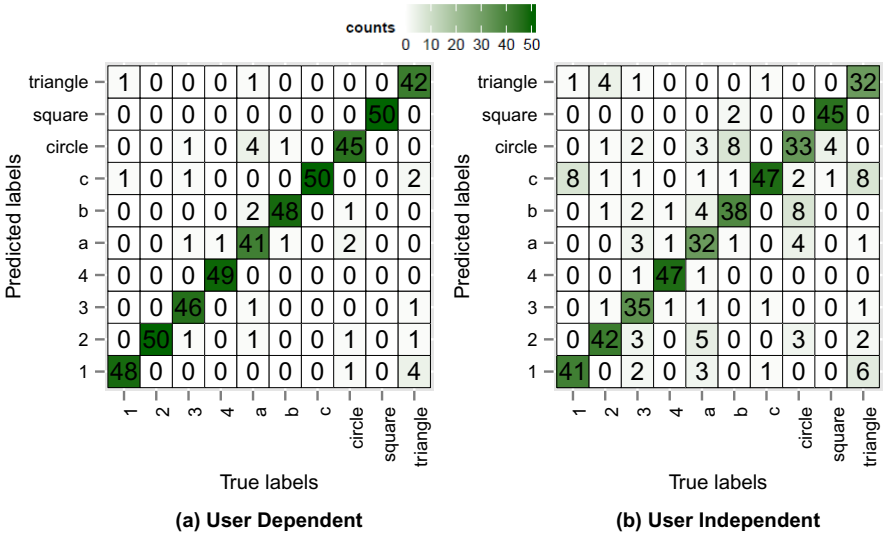

Fig. 5. Confusion matrices. a) user dependent case. b) user independent case.

example, Fig. 6 shows the layout of the third floor of an apartments building. It shows 3 of the 4 tested locations (The lobby is not shown. It is at the same level of bedroom A but in the first floor.) The locations were chosen to be very close to each other.

The experiments consisted of collecting the in range Access Points BSSID and signal strength. For each location the samples were collected for 3-4 minutes. The tests were performed using 10-fold cross validation on each of the scenarios. Table 2 shows the achieved accuracy and the number of locations for each scenario.

The expected combined accuracy (gesture + location) is the product of the two independent accuracies. Thus, for the user-dependent case it is 88.7% and for the user-independent case it is 74.2% however, more robust tests and experiments must be designed in order to conveniently evaluate the contribution of the context and the interactivity of the system in more scenarios which will be left as future work.

Table 2. Location recognition accuracy

Scenario	No. locations	Accuracy
Apartment 1	3	99.0%
Apartment 2	4	92.3%
House 1	5	93.7%
House 2	4	93.6%
	average	94.65%

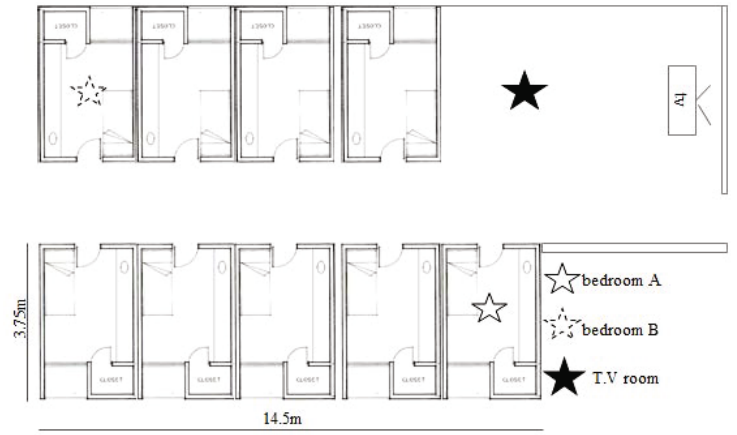

Fig. 6. Apartment 2 layout

8 Conclusions and Future Work

In this work it was shown how contextual information can be used along with the gesture recognition process to build a more user friendly and interactive system. The contextual information consisted of the user's location. A working prototype was implemented using a smartphone. Unlike other systems, all the computation is performed by the smartphone instead of delegating it to an external server. The main contribution of this work was that the gestures were recognized within a location context. This enables the system to be context-aware and thus more interactive and user friendly.

For future work we plan to include more information about the context, e.g, time of the day, temperature, historical data of the user's activities, etc. We also plan to use a probabilistic graphical model as inference engine instead of simple conditional rules.

Acknowledgements. Enrique would like to thank Juan Pablo García for his suggestions and feedback, and to Consejo Nacional de Ciencia y Tecnología (CONACYT) and the AAAmI research group at Tecnológico de Monterrey for the financial support in his PhD. studies.

References

1. Pebble, `http://getpebble.com/` (accessed March 22, 2014)
2. Jawbone up, `http://jawbone.com/up/` (accessed March 22, 2014)
3. Rabiner, L., Juang, B.-H.: Fundamentals of speech recognition. Prentice hall (1993)
4. Mace, D., Gao, W., Coskun, A.: Accelerometer-based hand gesture recognition using feature weighted naïve bayesian classifiers and dynamic time warping. In: Proceedings of the Companion Publication of the 2013 International Conference on Intelligent User Interfaces Companion, pp. 83–84. ACM (2013)

5. Wu, J., Pan, G., Zhang, D., Qi, G., Li, S.: Gesture recognition with a 3-D accelerometer. In: Zhang, D., Portmann, M., Tan, A.-H., Indulska, J. (eds.) UIC 2009. LNCS, vol. 5585, pp. 25–38. Springer, Heidelberg (2009)
6. Akl, A., Valaee, S.: Accelerometer-based gesture recognition via dynamic-time warping, affinity propagation, & compressive sensing. In: 2010 IEEE International Conference on Acoustics Speech and Signal Processing (ICASSP), pp. 2270–2273. IEEE (2010)
7. Pu, Q., Gupta, S., Gollakota, S., Patel, S.: Whole-home gesture recognition using wireless signals. In: Proceedings of the 19th Annual International Conference on Mobile Computing & Networking, pp. 27–38. ACM (2013)
8. Kühnel, C., Westermann, T., Hemmert, F., Kratz, S., Müller, A., Müller, S.: I'm home: Defining and evaluating a gesture set for smart-home control. International Journal of Human-Computer Studies 69(11), 693–704 (2011)
9. Carlotto, A., Parodi, M., Bonamico, C., Lavagetto, F., Valla, M.: Proximity classification for mobile devices using wi-fi environment similarity. In: Proceedings of the first ACM International Workshop on Mobile Entity Localization and Tracking in GPS-Less Environments, MELT 2008, pp. 43–48. ACM, New York (2008)
10. Salvador, S., Chan, P.: Toward accurate dynamic time warping in linear time and space. Intelligent Data Analysis 11(5), 561–580 (2007)
11. Krumm, J., Horvitz, E.: Locadio: Inferring motion and location from wi-fi signal strengths. In: First Annual International Conference on Mobile and Ubiquitous Systems: Networking and Services (Mobiquitous) (2004)
12. Correa, J., Katz, E., Collins, P., Griss, M.: Room-level wifi locationntracking. Carnegie Mellon Silicon Valley, CyLab Mobility Research Center technical report MRC-TR-2008-02 (2008)
13. Zdruba, G.V., Huber, M., Karnangar, F.A., Chlarntac, I.: Monte carlo sampling based in-home location tracking with minimal rf infrastructure requirements. In: Global Telecommunications Conference, GLOBECOM 2004, November 3-December, vol. 6, pp. 3624–3629. IEEE (2004)
14. Jaccard, P.: Nouvelles recherches sur la distribution florale. Bull. Soc. Vaud. Sci. Nat. (1908)
15. Witten, I.H., Frank, E., Hall, M.A.: Data Mining: Practical Machine Learning Tools and Techniques, 3rd edn. The Morgan Kaufmann Series in Data Management Systems. Elsevier Science (2011)