

Design and deployment of a contactless hand-shape identification system for smart spaces

Ana M. Bernardos¹ · Jose M. Sánchez¹ · Javier I. Portillo¹ · Xian Wang¹ ·
Juan A. Besada¹ · José R. Casar¹

Received: 5 October 2015 / Accepted: 10 January 2016 / Published online: 3 March 2016
© Springer-Verlag Berlin Heidelberg 2016

Abstract This paper explores the feasibility of a contactless identification system based on hand features. The identification solution has been designed to be integrated with smart space applications and relies on a commercial 3D sensor (i.e., Leap Motion) for palm features capture. The first part of the paper is devoted to evaluate the significance of the different hand features and the performance of a set of classification algorithms. 21 users have contributed to build a testing dataset; for each user, the morphology of each hand has been gathered from 52 features, which include bones length and width, palm characteristics and relative distances among fingers, palm center and wrist. In order to get consistent samples and guarantee the best performance for Leap Motion device, the data collection system provides sweet spot control. This functionality guides the user to situate the hand in the best position and orientation with respect to the device. The selected classification strategies—nearest neighbor (NN), supported vector machine, multilayer perceptron, logistic regression and tree algorithms—have been evaluated on Weka. We have found that relative hand-pose distances are more significant than pure morphological features. On this feature set, the highest correct classified instances (CCI) rate is reached through the multilayer perceptron algorithm (>96 %), although all the evaluated classifiers provide a CCI rate above 90 %. The analysis also gather how these algorithms perform with a variable number of users in the database, and what the sensitivity to the number of training samples is for each

algorithm. Results show that there are different alternatives that are accurate enough for non-critical, immediate response applications. The second part of the paper focuses on the implementation of application examples that are integrated with a real-time hand-based identification system using NN. In particular, the applications enable customization and gesture-based control of the smart space. A five-user study has provided insight into the system performance and user experience. Results confirm the viability of using in-air hand shape recognition in smart space settings, although also it is still needed to deal with aspects that hinder faster and more accurate recognition.

Keywords Biometry · Hand-shape based identification · Classification · Smart spaces · Gesture-based interaction

1 Introduction

Nowadays, there is an increasing availability of low-cost devices that are suitable to capture in-air gestures. Although these devices are often designed to interface with laptops or applications (e.g., Leap Motion), they have a great potential as interaction tools in our daily life environments (Tistarelli and Schouten 2011). In this direction, a lot of work about gesture recognition has been done. Solutions for gesture classification are usually categorized as user-independent or user-dependent; user-dependent algorithms need to know the user's identity to match the registered data with a previously stored personal template. In general, user-dependent algorithms provide better accuracies (Wang et al. 2015), even if the almost unavoidable additional training effort may reduce the system's usability. Our previous works in this area have led us

✉ Ana M. Bernardos
anamaria.bernardos@upm.es

¹ Universidad Politécnica de Madrid, ETSI Telecomunicación,
Av. Complutense 30, 28040 Madrid, Spain

to search for a comfortable identification mechanism that eases the input of the user's identity into the gesture recognition system.

In particular, the objective of this work is to explore the possibilities of delivering a functional identification method based on contactless hand shape analysis to enable non-critical services in smart environments; due to service needs, the target sensor is the Leap Motion device. This paper is an extension of (Bernardos et al. 2015); it includes an experimental comparison of performance for different classification algorithms regarding accuracy, training needs and scalability, and an analysis on usability and user experience for two real services, which have been built using a functional implementation of the contactless identification system. One of the services also integrates gesture-based control on Leap Motion, applying the gesture recognition algorithms described in (Wang et al. 2015). Although there is an increasing number of articles that rely on the use of this device for gesture recognition (Marin et al. 2015; Avola et al. 2014; Han and Gold 2014), to the best of our knowledge there is no prior work that specifically evaluates Leap Motion for hand shape-based identification.

The structure of the paper is as follows. Section 2 provides a review of previous research on shape hand-based identification. Section 3 defines the identification strategy itself, starting by the hand features to be used. Within this Section, it is also described the classification algorithms and the sweet pose tool used to gather the testing dataset. Section 4 presents a performance comparison that analyzes the relevance of the hand features considered; moreover, it evaluates some specific implementations of the algorithms, taking into consideration their accuracy, time to build the model, need for training and scalability. Section 5 explains the final technological choice to implement the identification system to be used in two smart space services, while Sect. 6 describes an introductory user study that has been carried out to evaluate the system in real operating conditions and the perceived user experience. Finally, Sect. 7 discusses on results and further work.

2 Related work

Hand-shape recognition is non intrusive and easy to operate by the user; additionally, the extraction of the hand shape information does not need very high resolution images. This fact facilitates the processing and storing needs, also providing better subjective acceptance than other biometric techniques, such as iris recognition or fingerprint recognition. This biometric technique relies on a sensor device (scanner, camera, Leap Motion, etc.), which takes a sample of the hand shape and, in many cases,

performs some kind of preprocessing (alignment, segmentation, etc.). Then, some feature set is extracted from the preprocessed hand image. Many hand-based biometric schemes work obtaining geometric measures of the hand and then extracting a set of features from these geometric measures. The main hand recognition approaches are based on hand geometry, hand contour and palm-print (Duta 2009). The first system to capture hand and finger images is dated in 1858 (Sodhi and Kaur 2003). In the mid 1960's, Robert Miller (1971) invented a mechanical hand geometry identification device. The first commercial device (Identimate) used mechanically scanned photocells to measure the finger length, the endpoint contours and the skin translucency. This device was in use from 1970's to 1987. In 1986, it was presented the ID3D HandKey (Jain et al. 2007), a device using low-cost digital imaging sensors. Currently, the increasing number of commercial systems and patents demonstrates the effectiveness of this biometric approach (Kong et al. 2009; Adán et al. 2008; Kumar and Zhang 2006).

Hand geometry-based systems use only hand geometric features, for instance, finger lengths, finger widths, aspect ratio of the palm or the fingers, hand length, thickness, hand area, palm area, measure ratios, etc. The number of features varies from 13 to 40. These methods reduce the information given in a hand sample to a N -dimensional vector that is used to implement a matching algorithm based in a metric distance, such as Euclidean distance (Sanchez-Reillo et al. 2000), Mahalanobis distance (Jain and Duta 1999), absolute (L1) distance (Yörük et al. 2006), correlation coefficient (Kumar and Zhang 2006), or some combination of these distances (Pavešić et al. 2004). Other alternative schemes are proposed in literature applying different probabilistic and machine learning techniques, like k -nearest neighbors (Kumar and Zhang 2007), Gaussian mixture models (Wong and Shi 2002), or support vector machines (Kumar and Zhang 2006, 2007; Yuan and Barner 2006; Guo et al. 2011). For instance, in Morales et al. (2008), 40 features obtained from finger widths for 3 fingers are used to train a Supported Vector Machine (SVM). In Adán et al. 2008, a hand natural reference system is used to make the system robust against different hand poses, and the classification is based in a time averaged feature vector. This system is based on a webcam. Sánchez-Reillo et al. (2000) use 25 features, such as finger widths, finger and palm heights, finger deviations and angles of the interfinger valleys with respect to the horizontal, using Gaussian mixtures to model them. Ross and Jain (1999) use an imaging scheme to select 16 features, such as the length and width of the fingers, the aspect ratio of the palm to the fingers, and the thickness of the hand. Oden et al. (2003) use geometric features and finger shapes. The finger shapes are modeled using fourth degree

polynomials. They obtain 16 features that are compared using the Mahalanobis distance.

Hand contour based systems use the hand silhouette to extract the features and to perform the matching. For instance, Yörük et al. (2006) use 2048 points of contour coordinates to construct a raw feature vector and independent component analysis features are used in the identification and verification tasks. Woodard and Flynn (2005) use shape indices based on 3D shape curvature and a match score based in correlation coefficients between shape descriptors.

Palm-print systems use the silhouette lines for matching, frequently in combination with geometric measures. For instance, Kanhangad et al. (2011) use a 3D digitizer to extract intensity and range images from the user hand and then, multimodal palm print and hand geometry features are obtained for matching. Kumar et al. (2003) describe a bimodal biometric system using hand geometry and palm print information and propose a strategy to fuse both sources.

It is also important to present the evolution and perspective of the hand shape recognition systems from the operation point of view, specifically, the sensor (image acquisition system). The early hand recognition systems were contact based, using a platform and pegs or guides to help the user to situate the hand (Sanchez-Reillo et al. 2000; Jain and Duta 1999). The next generation was composed of unconstrained systems (without mechanical aids), but yet with the necessity of placing the hand on a platform or scanner (Adán et al. 2008; Ferrer et al. 2009). Modern hand shape identification systems (such as the Leap Motion, the one considered in this paper) are unconstrained and contact free (Zheng et al. 2007; de Santos-Sierra et al. 2014), with mild restrictions on the user hand situation. The hand position is almost free and there are no platforms or scanners to situate the user hand. Contactless hand recognition systems are increasingly receiving attention because of their better user acceptability, and their capability to be extended to daily devices such as smartphones. The feasibility of hand recognition using low-cost devices is also an issue to consider. For instance, de Santos-Sierra et al. (2014) present an algorithm to segment hand images using multilayer graphs and Mostayed et al. (2009) use low resolution hand images and compute a set of position invariant features using the Radon transform.

Our approach in this paper is to use a contact free (in-air) device (the Leap Motion) to retrieve the hand palm features. On the collected dataset, we will analyze the meaningfulness of the features and apply different classifiers to explore the feasibility of building an effective identification system for smart spaces applications.

3 Defining a hand shape-based identification strategy

The service scenario that we are envisioning in this work considers non-critical applications that can benefit from the availability of a straight, fast and usable identification process (refer to Sect. 5 for some real examples). Let's think on a user that is back at home and wants to keep on watching the film he did not finished the evening before. He sits on the sofa; the sofa arm has an integrated Leap-motion like device to enable room control. Thus just by waving the hand on the sofa arm, "the room" is able to recognize who the user is and, by a subsequent gesture chain, it can interpret what the user wants to do. With this idea in mind, our identification system needs to:

- *Provide contactless hand shape-based identification* The identification capability will be integrated in an in-air gesture recognition system, so the identification has to be done on in-air input.
- *Be accurate enough for non-critical applications* The applications to deploy on top of the identification system are related to interaction and personalization of smart environments. Being not critical applications, the global performance of the identification system undoubtedly conditions the user experience.
- *Be easy to use* The identification process has to be as simple as possible, providing enough feedback cues to the user in order for him to control the interaction.
- *Work with minimum training* Supervised algorithms require the users to train them to work. Although the training stage may be unavoidable to reach sufficient accuracy, the identification method must reduce it to a minimum.
- *Provide real-time response* The identification stage must be as quick as possible, providing immediate feedback.
- *Be robust* The solution has to work coherently under different scenarios with external variable conditions. For example, the illumination in the environment can change.
- *Be scalable for smart environment-like scenarios* The solution has to be validated with a number of users that is considered reasonable for medium-sized spaces (a smart home, a small office, etc.).

Our objective is to define a solution (device, hand features and classification algorithms) that fulfills the requirements above.

3.1 Device, hand features and algorithms

The technological sensor choice to implement the identification system has been the Leap Motion sensor. Leap

Motion, developed by the same named company, is a small USB peripheral device that supports hand and finger motions as input, not requiring hand contact or touching. Launched to market in 2013, it was initially conceived to interact with a computer. Two monochromatic infrared (IR) cameras and three infrared LEDs, which generate a 3D dot pattern of IR light, compose the sensor. From the comparison of the 2D frames provided by the two cameras, dedicated software in the computer synthesizes the 3D position data of the hand. The coordinate system used by this device is a Cartesian coordinate system with its origin placed in the Leap Motion's center (Fig. 1a).

The sensor can reach up to 200 frames per second in the best conditions. Every frame delivers information about the hands by comparing the IR scenes against an internal hand model. The model defines that each hand has five fingers formed by four bones (metacarpal, proximal, intermediate and distal phalanx) except for the thumb, which is formed by three (proximal, intermediate and distal phalanx). In this work, we take benefit from the API provided by Leap Motion, which can recognize hands, fingers, arms and tools (straight cylindrical objects longer and with a smaller radius fingers) over it. For each finger and bone, the API provides its width and lengths. Furthermore, it is possible to get information of the palm and wrist width, the palm orientation or the point direction for each finger. These libraries also allow recognizing four predefined gestures (swipe, key tap, screen tap and circle), provide the images acquired by the two cameras and discern between the two hands that the user may be using. Figure 1b and c shows a simulation of the hand view.

Leap Motion's API directly provides sufficient geometrical hand features that can be used for identification. Taking into consideration existing literature (Sect. 2), the classification strategies detailed below will initially work on 52 geometric hand features: (a) intrinsic morphological hand features, i.e., finger (5 features) and phalanx lengths (19) and finger, palm and wrist widths (7) (Fig. 1b) and (b) pose hand features, i.e., intra-hand distances between

the fingertips (10), from the palm to the fingertips (5), from the wrist to the fingertips (5) and from the wrist center to the palm center (1) (Fig. 1c). The use of intra-hand distances assumes that the user is systematic in the way s/he extends the hand over the sensor, as it relies on the relative distances between difference reference points within the hand (fingertips, palm center, wrist center).

On these features, we aim at analyzing different classification techniques to choose the most suitable one for our purposes. The supervised classification process includes a training phase in which a set of n pattern feature vectors $\{x_1, x_2, \dots, x_n\}$ (in our case, the features described in the previous section) are gathered, assigning them to the n classes to classify (in our case, the users' identities), finally building the classification model. After taking a real-time sample, a feature vector y is computed. The vector y is then compared to the pattern feature vectors using different strategies and the class that is identified to be the 'nearest' to the feature vector y is given as result. From a review of the existing literature, we have decided to study the performance of well-know classification algorithms, such as nearest neighbor, neuronal networks (multilayer perceptron), support vector machine (SVM), Logistic Regression and tree-based algorithms (such as functional trees or logistic trees) for our classification problem. We expect some of these methods to provide good or very good accuracy, while delivering different performance in terms of model building and classification time (computational cost).

3.2 Dataset description and test environment

We have gathered a dataset of hands snapshots using Leap Motion; it contains data from 21 users—18 males and 3 females, with ages between 23 and 53. A hand-image recording tool has been designed and implemented to enable feature recording (Fig. 2). For each user, 40 snapshots of both right and left hands have been recorded (80 samples \times 21 users in total); within each snapshot, 52 hand features have been collected (89.040 features). It has

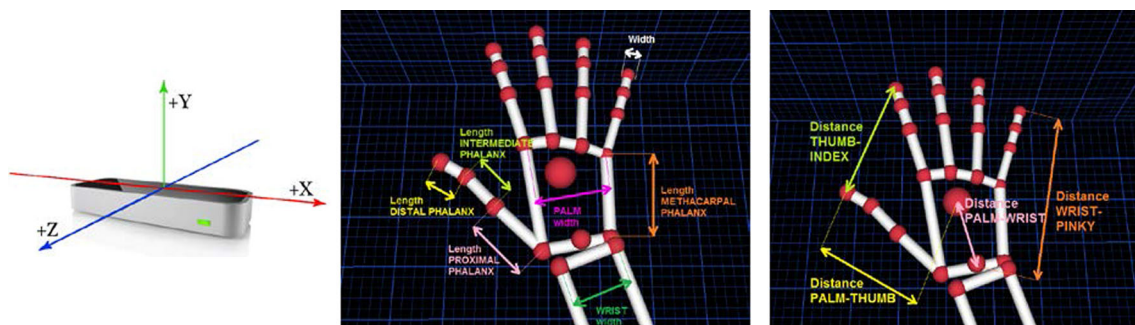


Fig. 1 a Leap Motion and its axis; b Hand length-width based features; c Hand distance features

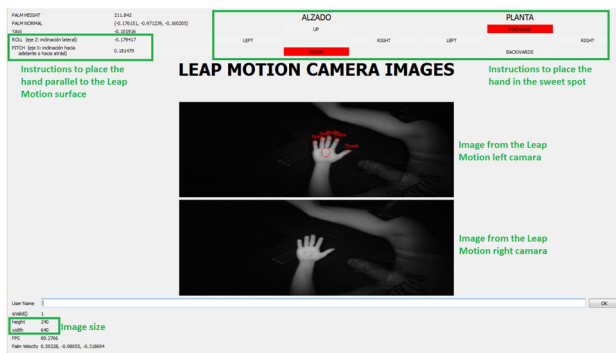


Fig. 2 Training application. In this case, the user has to move the hand down and forward to fulfill the sweet spot indications

taken 20–25 min for each user to provide the 80 required samples.

The interface in Fig. 2 provides real-time feedback to help the user to situate the hand in the optimal pose to record the hand features, in terms of position and orientation; this pose is usually referred as ‘sweet spot’. The interface is configured to provide real time indications for the user to place the hand at 18–21 cm over the Leap Motion device (y axis), in the middle of it (x axis between -1.5 and 1.5 cm) and slightly advanced with respect to the device (z axis between -3 and 7 cm). The sweet spot guarantees the best vision of the hand, as the most accurate measurements are obtained when the hand is between 15 and 25 cm and in the negative values of the z -axis (Guna et al. 2014). The indications to the user are shown through red tags that are activated in case the hand is not in the perfect position; the positions of the tags in the screen indicate the direction to move the hand. The user will also receive feedback to correct the pose if the hand is not sufficiently parallel to the device. The interface includes information that is relevant for the developer (e.g., the camera views), but that will not be part of the real time system. When the user’s hand is correctly situated, the system takes a snapshot. The user has to take the hand out from the device’s line of sight and situate it correctly again, for the next snapshot. Users rapidly get their own references to correctly place the hand, so the process is much faster as the user takes practice.

4 Performance analysis and comparison

The final dataset has been post-processed to serve as input for the well-known weka open source data mining software. The use of weka facilitates doing fast prototyping of different classification solutions, in order to know in advance about their e.g., accuracy, time to build the model, feature relevance and training sensitivity. All algorithms

that are to be evaluated have a configurable Weka implementation available (Table 1 shows some of the configuration parameters that have been applied for each algorithm).

For the experiments, Weka 3.6.11 ran in a HP Z1 Workstation (CPU 3.3 GHz, RAM 8 GB). Table 2 gathers the percentage of correct classified instances (CCI) applying 10-fold cross validation on the right and left hand datasets (Leap Motion correctly estimates the type of hand even if it is turned up). Results are calculated using the information from (a) all gathered features (52 features), (b) those containing the distance attributes (21) or (c) a subset of this latter group (11). This subset has been build from a Weka analysis on the most meaningful attributes. Three different implementations of the Best-First search algorithm (BestFirst, GreedyStepwise and LinearForwardSelection) have selected 11 distances as most meaningful features: palm-thumb, palm-pinky, wrist-pinky, thumb-index, thumb-ring, index-middle, index-ring, index-pinky, middle-ring and middle-pinky. Rank-Search adds thumb-middle to the list. And the Ranker method undoubtedly ranks the distance-based features as the most meaningful for the classification. Thus distance-based features seem to contain more information globally.

As the reader will notice, the use of distance-based features maintains or even increases the percentage of CCI. For example, in the case of IB1, the CCI rate increases 2.4 points in the right hand case and 2.6 points for the left hand. Nearest Neighbor (NNge type) and Sequential Minimal Optimization (SMO) are also positively affected (in the case of the right hand, the CCI rates are improved 1.78 and 1.19 points). The rest of methods show very slight performance variations. Results are similar when using 11 features instead of 21.

Let’s consider the right hand dataset (most potential users will be right-handed, as the percentage of left-handed in the world population is between 8–13 %). In this case, the algorithms that perform better (>96 %) both for 52 and 21 features are multilayer perceptron (MP), logistic and the logistic model tree (LMT). Simple strategies, such as nearest neighbor (IB1), reach a reasonable 94 % accuracy that is preserved when reducing the number of features to 11.

With respect to the necessary time to build the model (the model has to be re-calculated each time that a new user is included in the dataset), MP, LMT, Logistic and Logistic Model Tree are the slower strategies. The model-building time is naturally reduced with the number of features to consider in the classification process. In the case of the right hand dataset, the time to build the model is reduced up to 81 % for the Logistic method, 62 % for Multilayer Perceptron and around 50 % for Simple Logistic and Tree-based methods when moving from 52 to 21 features. The

Table 1 a Weka available implementations for the selected algorithms. **b** Main configuration parameters for the classifiers. More information available at <http://www.cs.waikato.ac.nz/ml/weka/>

Classification algorithm (weka type)		Weka available implementations						
Nearest-neighbor (NN, rule)		NNge, nearest-neighbor-like algorithm using non-nested generalized exemplars						
Nearest-neighbor (NN, lazy)		IB1. Uses euclidean distance						
Multilayer perceptron (MP, func.)		Uses backpropagation to classify instances						
Sequential minimal optimization (SMO, functions)		John Platt's sequential minimal optimization algorithm for training a support vector machine classifier						
Logistic (functions)		Class for building and using a multinomial logistic regression model with a ridge estimator						
Simple logistic (functions)		Class for building linear logistic regression models. LogitBoost with simple regression functions as base learners is used for fitting the logistic models						
Functional (FT, trees)		Classification trees that could have logistic regression functions at the inner nodes and/or leaves						
Logistic model (LMT, trees)		Class. trees with logistic regression functions at the leaves						

Class.algorithm		Classifier parameters							
NNge	No. attempts for general. = 5	No. folder for mutual info. = 5							
Multi. perceptron	Hidden layers = 1	Learning rate = 0.3	Momentum = 0.2	Seed = 0	Training T. = 500	Validation set size = 0	Validation threshold = 20		
SMO	c = 1	Epsilon = 10^{-12}	Filter type: normalize training data	Kernel: PolyKernel-C 250007 -E 1.0	Num. Folds = - 1	Random seed = 1	Tolerance parameter = 0.001		
Logistic	Max. iterat. = -1	Ridge = 10^{-8}							
Simple logistic	Heuristic stop = 50		Max. boosting iterat. = 0		Weight trim beta = 0				
FT	Min. no. instances = 15		Model type = FT	No. boost. iter. = 15	Weight trim beta = 0				
LMT	Min. no. instances = 15		No. boosting iterations = -1		Weight trim beta = 0				

Table 2 Correct classified instances (%) and time to build the model on 52, 21 and 11 input features. Computing time is provided for the right hand dataset

	52 Features			21 Features			11 Features		
	Right hand	Left hand	Time (s, right hand)	Right hand	Left hand	Time (s)	Right hand	Left hand	Time
NNge	89.05	90.36	0.35	90.83	92.38	0.22	90.83	91.55	0.14
IB1	92.26	90.48	0.01	94.64	93.1	0.01	95.12	93.93	0.01
Multi. Perce.	96.67	95	34.91	96.79	95.71	13.22	95.60	94.26	9.12
SMO	91.07	89.05	0.56	92.26	91.55	0.53	90.71	91.31	3.24
Logistic	96.43	94.64	16.59	96.31	95.12	3.09	93.93	93.57	3.15
Simple Log.	96.43	94.76	13.69	94.43	94.88	6.77	96.07	93.81	5.05
FT	95.12	93.93	5.17	95	93.93	2.49	93.93	92.98	1.64
LMT	96.43	95	44.08	96.43	94.88	21.58	96.07	93.81	31.48

IB1 (NN) method is the only one that remains unaffected, due to the very small time it needs to set its model up.

An important issue to build a real-time system is to have a reference on the minimum number of training samples

that are needed to get a reasonable accuracy. The length and complexity of the training directly affects the user experience (obviously users are not willing to train the systems, not even once if possible). Figure 3a shows how

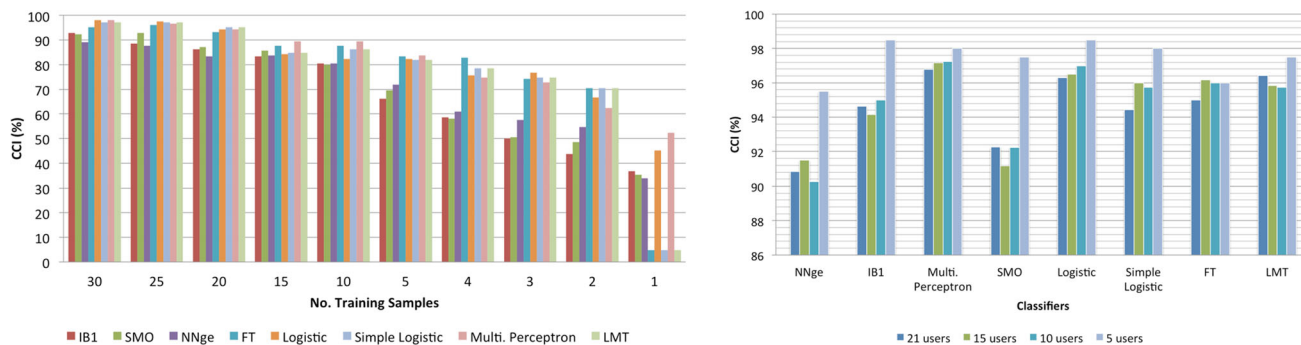


Fig. 3 Classifiers performance taking into consideration: **a** the number of training samples and **b** the users in the database

the different algorithms perform with a decreasing number of samples. The experiment has been configured to integrate an increasing number of training samples and, the test dataset of ten samples, to check the performance. The 21-features right hand dataset has been used. Globally, logistic methods, multilayer perceptron and trees are more robust to the decreasing number of training samples, while NN-methods and SVM algorithms reduce their accuracy significantly. Training with a single sample provides not sufficiently good results, but there is a significant change when 2–3 samples are used. When applying the first group of algorithms, at least five trainings are needed to reach a CCI rate above 80 %. For the second group of algorithms, ten samples are required to reach this same percentage of CCI. With respect to the necessary time to build the model, the slower method is LMT (50.17 s for 30 training samples), followed by the multilayer perceptron (25.6 s). The fastest are the NN methods (<1 s).

Finally, it is also important to know how the different algorithms perform with respect to the number of users in the dataset. Scalability is key for many identification systems. In this case, our design requirements need that the identification system works well on a not too large number of users. Figure 3b gathers some results in this direction. Experiments have been carried out on the right-hand 21 features dataset, all samples, and 10-fold cross validation. Every algorithm achieves its better performance with the smallest number of users (5), except Functional Trees (FT) that provides a very small variation of the CCI rate (+0.1 %). Some algorithms perform more stably than others. It is the case of Multilayer Perceptron or the Trees, which CCI rate is around 1 % lower when comparing the 5-user dataset and the 21-user dataset. On the other hand, SMO losses 5.2 %, NNge 4.7 %, IB1 3.9 %, Logistic 2.2 % and Simple Logistic 3.6 % when comparing these two situations. Some other algorithms are very sensitive to the change from 5 to 10 users and provide very small variations on the CCI rate in the remaining cases. It is the case of NN methods and SVM.

5 Building real services on top of the identification system

After the previous analysis, we have move forward with the implementation of a real time hand-based identification system. Our choice to implement a solution delivering real-time Leap-based identification has been to use the NN algorithm, which provides a reasonable performance in terms of accuracy and recognition time for our purposes. Using the real time libraries for identification, we have built two services that may be deployed in a smart space; these services make use of the identification capabilities to facilitate space customization and control. More details on the services and the user interfaces are included below.

5.1 Service description and interfaces

The identification system enables the user to train the system with the hand palm. Figure 4a shows the user interface, with the sweet spot indications to help the user to situate the hand in the best way to store the palm pattern. These sweet spot indications will facilitate the user to place the hand in a similar way in real-time operation.

The first service that has been implemented on top of the identification system is an Automatic Space Customization Service, which enables to configure the preferred settings for a smart room (e.g., light colors, music, blind positions, etc.). On identification, the space is automatically configured with the previously stored preferences for the identified user. Figure 4b shows the interface that facilitates storing the preferences for space customization.

The second service is about space control. In this case, an in-air gesture-based recognition system has been implemented on Leap, applying the dynamic time warping (DTW) algorithm that has been proposed in our previous work (Wang et al. 2015). The gesture-based recognition system works as follows: the user trains some letters or directional gestures by repeating them several times, and the system builds a template for each gesture and stores it,



Fig. 4 **a** Identification interface. The figure is tagged with (1) sweet spot indications—the arrows indicate the direction in which the user has to move the hand; (2) recognition feedback; (3) link to the training interface and (4) link to reset the workflow. **b** Interface for space customization. The figure is tagged with different numbers: (1) The

user who is configuring the space. (2) The room being configured (the displayed resources to configure are filtered taking location into consideration). (3) Video genre. (4) Blinds configuration. (5) Lighting configuration (Philips Hues in the chosen room)

for it to be used in real time. In particular, we have built a compiler that enables the recognition of a control grammar composed by 2–3 gestures. In this grammar, the first letter identifies the target object by its initial and the second gesture identifies the action to perform. In some cases, a third object is involved in the action, which is also identified by its initial letter. For example: if the user wants to put a blind down, s/he will draw in the air a *b* (for blind) and a *downwards* movement to indicate the action. In this case, the identification system is used to compare the real-time input with the user's template to discern which gesture s/he is performing. It is out of the scope of this paper to describe in depth the gesture recognition algorithm and analyze its performance, but interested readers may refer to Wang et al. (2015) for additional information. Figure 5a shows the gesture training interface, while Fig. 5b shows the interface for the real-time control part. The interfaces provide feedback on the way that the gesture is drawn and also sweet spot indications to facilitate the similarity of the movements to the user's templates.

5.2 Experimental setting and implementation details

These services have been deployed in the Experience Lab of Future Spaces, a 160 m² laboratory that has been equipped to develop user experience tests in realistic daily living environments (home, office, public space-like). The high level architecture for the deployment is depicted in Fig. 6. The ExpLab provides a tailored smart home infrastructure that enables to receive events, sensing and controlling objects such as lights, blinds, robots, etc., through easy-to-use APIs (Smart Home Manager). Additionally, a DLNA-based infrastructure (DLNA stands for digital living network alliance) facilitates the management of media contents around the space, among TVs, mobile devices and projectors. The DLNA infrastructure is

composed by a media server (for the contents), the renderers (standard for many devices) and a controller (that handles the content transfer between devices). The controller provides an API which facilitates the whole control of the media effect. More information about the DLNA infrastructure can be found at Bergesio et al. (2013).

Additionally, there is a centralized infrastructure to manage the services in the Laboratory. This infrastructure relies on a database, in which every element of the system is registered. In particular, the services described above (space customization and space control services) use this centralized database to store the users of the system and their preferences about the media contents and smart home elements, depending on the room.

With respect to the implementation details, the approximate nearest neighbor library (ANN) library¹ has been integrated in the C++ code that contains the core hand-shape based identification system (hand recognition block). Qt has been the chosen language for interface design, class communication and thread creation.

The code structure includes two main libraries:

- *LeapMotionHandsLib*, which enables reusing the ANN identification implementation in any project. Two main classes compose this library: *HandID*, for identification and *HandTraining*, to manage the training process with its own interface.
- *LeapMotionGesturesLib*, composed by a class containing the implementation for the Dynamic Time Warping algorithm (*eDTW*) and two classes that enables gesture recognition: *GestureID*, using the DTW algorithm for gesture identification and *GestureTraining*, which enables to train the gesture patterns through a specific interface.

Then, other relevant classes are:

¹ Available at: <http://www.cs.umd.edu/~mount/ANN/>.

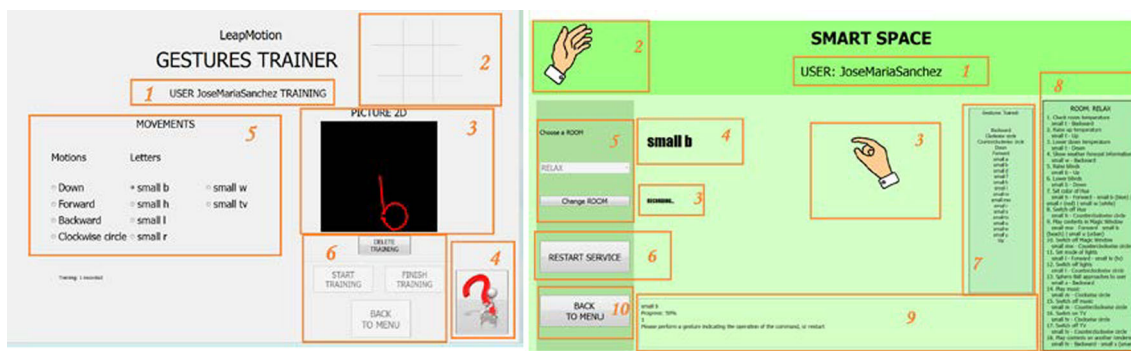


Fig. 5 a Interface for in-air gesture training. The tags refer to (1) the user’s id; (2) the matrix to show whether the gesture is being performed in the optimal field of view (if not, red arrows indicating the direction to move are shown); (3) 2D blackboard showing the trait (to check if it is similar to the gesture to perform); (4) help option; (5) gestures that can be trained and (6) management options. **b** Interface

for gesture-based space control. The figure is tagged with (1) the user’s name; (2) hand detected and visible in the sensor range; (3) hand drawing a gesture; (4) next gesture to draw; (5) room to control; (6) button to restart the service; (7) list of trained gestures; (8) list of available grammar chains in the chosen room; (9) feedback on the grammar status; (10) button to go back to the menu

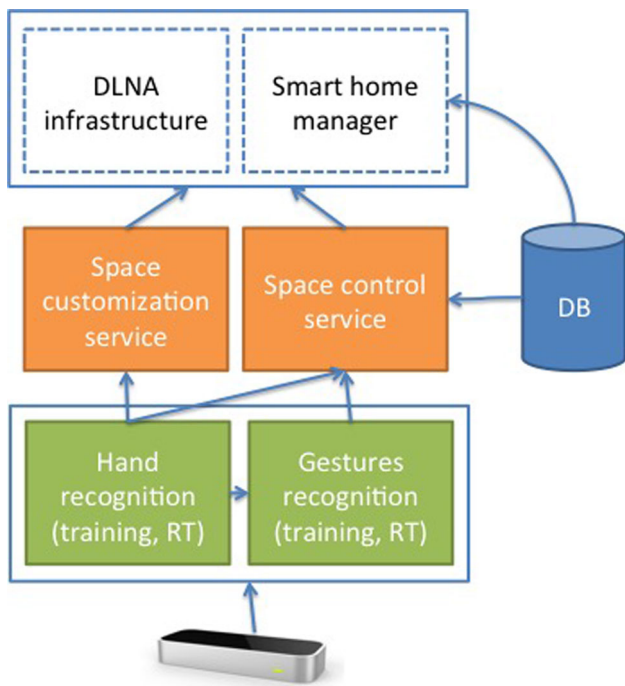


Fig. 6 High-level architecture

- *LeapManager* it handles the communication with the Leap Motion SDK, managing all the information coming from the sensor.
- *Start* it manages the configuration parameters for the system (security threshold and gesture duration).
- *IDTest* it contains the interface for user identification, providing the possibility of using or training the system.
- *AppMenu* the interface for the main menu, which includes the options of gesture training, space customization, interaction and control.
- *Customization* it handles the interface to configure the preferences for the scene.

- *SmartSpace* it enables gesture-based control.
- *DBWriter* and *Configuration* are classes interacting with the database to retrieve the users’ preferences and other types of connections.
- *MySocket* connects with the external gesture grammar compiler.
- *PreferenceManager* it is used to configure the environment when a user is recognized and the initial scene is configured by default. Its main class is *PutManager*, which connects with the external smart home controller and the DLNA infrastructure and encapsulates their responses in JSON objects.

The deployment of these two services has enabled to perform a short user study, which is described below.

6 User trial

The objective of our user trial has been, on one hand, to evaluate the performance of the recognition system in real-time operation and, on the other hand, to get some feedback on the user experience while using the palm identification mechanism in a realistic setting.

6.1 Experiment setting and tasks

Users have been recruited among university students and researchers (not involved in the project, in a range of age from 22 to 53). Five users have finally participated in the testing. As it is claimed in existing literature (Nielsen and Landauer 1993), this limited number of participants should be enough to collect insights on the main usability aspects, while identifying relevant design hindrances that may distort user experience.

Table 3 List of tasks

Test structure

Service 1. Identification and smart space customization

- T1. Train the system with your hand. (5 repetitions)
- T2. Identify yourself 12 times (4 times with each security level)
- T3. Personalize your preferred actions in the smart space through the options in the menu
- T4. Identify yourself to customize the environment with your preferences

Service 2. Gesture recognition and smart space control

- T5. Identify yourself to access the menu application and access the gestures training menu
- T6. Train the letters h, b and w and the directional gestures downwards, counter-clockwise circle and backwards, which will serve to use the smart devices. Take into account the guidelines in the screen
- T7. Now switch the HUE light bulb off by identifying the object with its initial letter (H) and then performing the action gesture (counter-clockwise circle)
- T8. Pull the blind down (gesture b + downwards)
- T9. Show the weather forecast on the screen (gesture w + backward)

The study has been structured to validate contactless hand-based identification and gesture recognition with Leap Motion, both features integrated in the services referred in the previous Section. For this reason, the session has been structured into two different parts: (a) identification and smart space customization, in which the hand-based identification system is trained and then evaluated in the framework of the space customization service (four tasks) and (b) gesture recognition and smart space control, also including gesture training and evaluation in the framework of the control service (six tasks). The task list is available at Table 3. The approximate duration of the test is around 60 min.

Regarding the test dynamics, prior to initiate it, the facilitator explains the user which the objectives of the session are and provides a profile questionnaire (on technology experience with kinect and leap devices and smartphones), the tasks to complete with questions regarding the task difficulty, and a final questionnaire on the services' usability and user experience, which ends up with open questions and suggestions to enhance the system.

Testers are also asked for authorization to be recorded in the informed consent form. During the trials, the facilitator is in charge of registering errors and annotating relevant aspects of the evaluation. The tests have been carried out in September 2015 in the previously mentioned Experience Lab (Fig. 7).

6.2 Results

With respect to difficulty, all tasks are considered as easy to perform (the average rating of difficulty is below 3) (Table 4). Training tasks are slightly more complicated (T1 and T6). The consecutive use of the hand-based identification system (T2) is also considered as slightly more difficult by some users. With respect to the gesture-based control, the difficulty seems to decrease with the number of interactions. All users have been capable of completing all tasks, with the exception of T5.

Results on identification performance (considering data gathered in T2) are shown at Fig. 8. Several thresholds can be configured for security reasons; in practice this means

Table 4 Tasks difficulty, rated by 5 users

Task	Task difficulty (1-very easy, 5-very difficult)		% Of completed tasks
	Mean	Std.	
T1	2.70	0.84	100
T2	2.70	1.22	100
T3	2.17	1.14	100
T4	1.51	0.54	100
T5	1.78	1	85
T6	2.93	0.70	100
T7	2.30	0.89	100
T8	1.74	0.44	100
T9	1.50	0.55	100



Fig. 7 The Space Customization service in action. When a user is identified, his preferred media contents and lighting configurations are set. The Leap Motion device can be integrated in everyday objects (e.g., on a sofa arm)

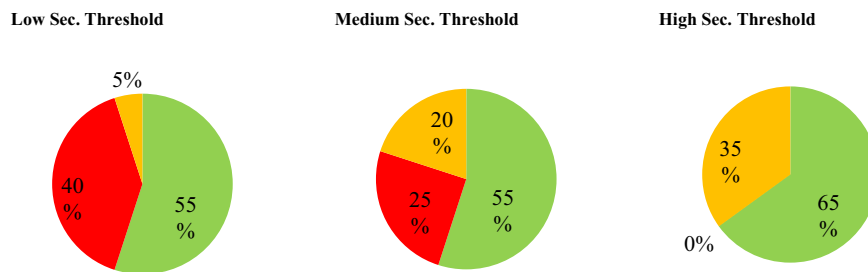


Fig. 8 CCI rate considering three levels of security and access times. Color code correct identifications (green), wrong identifications (red), failed attempts (yellow)

that the required distance between the real-time samples and the stored patterns can be adjusted to minimize false positives (wrong identified users) that grant undesired access to the service/system. The maximum time allowed for identification has been set to 60 s. Beyond that time, it is considered that the system fails to recognize the user. For each security threshold, each user has performed 4 recognition attempts (20 samples for each case). The number of trained templates in the system's memory remains constant: templates for 21 users are loaded in the database previously to any new training.

From the gathered data, it can be noted that the high security level does not provide any false positive. With the high security threshold, the recognition time increases, thus for non-critical standard services the low recognition threshold is preferred. Considering all the identifications that have been carried out with this threshold along the test (35, 7 per user in T1, T2, T4 and T5), a CCI rate of 67 % is obtained and wrong identifications rise up to 29 %. Although the percentages are not as good as expected, it is important to note that the use of Leap sensor usually needs some learning. Learning effects have been intended to be minimized with the use of sweet spot guidelines, but it is

still perceptible that after a few iterations, users learn how to position their hand in the way they had previously trained the system and the identification is faster. With respect to the recognition times, an average of 14.4 s ($\sigma = 8.3$ s) are needed to complete recognition with low security threshold. Medium and high security thresholds require 22.2 s ($\sigma = 12.2$ s) and 38.4 s ($\sigma = 16.5$ s) respectively. As the standard deviation shows, the variability of the time is quite high in any case.

Although it is not the core objective of the paper, Table 5 shows the training and recognition rates for different gestures. In general, users needed to learn how to correctly train the gestures (four users perceived that their skills improved with the number of iterations). In general, users were satisfied with the indications in the interface, one suggested that a 3D blackboard could be included. The backwards gesture was uncomfortable for four users. Users had to train each gesture five times, being this number in the upper limit considered as reasonable (from 2 to 5 trainings); one user told that 1-gesture training would be better. Regarding the CCI rate, 90 % of gestures in a 30 gestures sample were correctly recognized. It has to be noted that these data are calculated with a single try per

Table 5 Correctly trained gestures in a first attempt (generating a valid template) and CCI rate for real-time operation

Trained gesture	Correctly trained gestures (%)	CCI rate (%)
Small h	75	65.21
Small b	83	100
Small w	100	100
Up	88	100
Down	94	100
Backwards	71	88.23

user per gesture, as users were not asked to repeat the gesture specifically for recognition, but only to control the devices. As it can be seen, the “h” is the most problematic letter in the vocabulary, the same that the backwards gesture.

When taking into consideration the user’s expertise, it seems that users with previous experience with Leap Motion achieve a higher recognition rate in the hand-based identification, but there is no direct relationship between previous experience with the device and the quality of the gestures (Table 6).

After T4 and T9, users were asked to complete some qualitative questions to get some information about their feelings when using the services. Regarding the Space Customization service, users stated that they liked the hand-based identification functionality (5.3 in average, in a scale: 1-I did not like at all vs. 7- I loved it). Four users said that the concept was very comfortable for services requiring identification. Three users stated that they prefer hand geometry to passwords for identification. Additionally, three users had experienced other identification systems (e.g., voice, face geometry) and two of them preferred the Leap-based one. The preferred space to use this system is when at home (4 over 5 users would like to use the system in this environment). With respect to the gesture-based control, users enjoyed the system (5.9 over 7 in average) and stated their will of using the system at home or work. One of the users remarked that the system could be

interesting for impaired persons and two users underlined that the system was a bit slow. Mobile touch interfaces are still preferred over in-air gesture control.

Additionally, user experience-related questions have been included after the completion of each service tasks. User experience is “a consequence of a user’s internal state (expectation, needs, motivation, etc.), the characteristics of the designed system (complexity, usability, functionality, etc.) and the context within which the interaction occurs” (Hassenzahl and Tractinsky 2006). In our case, we have chosen the User Experience Questionnaire (UEQ) (Laugwitz et al. 2008) to measure the user experience. The questionnaire is composed of 6 factors (attractiveness, efficiency, perspicuity, dependability, stimulation and novelty) with 26 elements in total. Each of these elements belongs to a factor. Each factor is composed by two adjectives with opposite meaning. Between them, there is a 1–7 scale, so the user has to select to which adjective s/he feels the system/service is closer. The elements try to collect information to answer the following questions:

- *Attractiveness* General impression about the product, are the users enjoying the product?
- *Efficiency* Is it possible to use the product quickly and efficiently? Is the user interface neat?
- *Perspicuity* Is it easy to understand how to use it? Is it easy to get familiar with it?
- *Dependability* Has the user control over what’s happening? Is the interaction safe and predictable?
- *Stimulation* Is the product interesting to use? Does the user get motivation enough to use the product in the future?
- *Novelty* Is the product innovative and creative? Does it get the attention from the users?

Figure 9 summarizes the results obtained for both services. Regarding the Space Customization Service, the evaluation is reasonably positive. The weakest aspect of the system is its efficiency (1.25 points). The strongest aspects of the system are its attractiveness (1.77) and novelty

Table 6 Data referred to users’ expertise

User	Previous use of leap motion	Previous use of kinect	% Correct access			Re-trained letters	Gestures CCI rate
			Low Sec.T.	Med Sec.T.	High Sec.T.		
1	Sometimes	Often	100	100	100	h	85.7
2	Sometimes	Sometimes	100	100	100	b, h	81.8
3	Rarely	Sometimes	57	50	75	no	100
4	Rarely	Often	57	0	0	no	94.74
5	Rarely	Sometimes	29	25	50	no	90

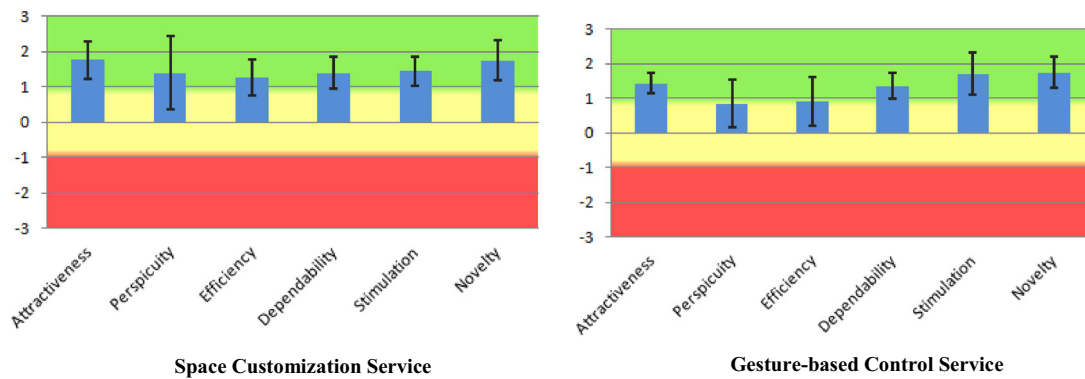


Fig. 9 UEQ results ordered by factors for space customization and gesture-based control services. *Rating* [−0.8 to 0.8]—neutral; [−0.8 to −1.5]—negative; <−1.5—very negative; [0.8 to 1.5]—positive; >1.5—very positive

(1.75). With respect to the space control service, conclusions are slightly worse than for the previous service, although still positive. The weakest aspect is perspicuity (how easy the product is to understand or to get familiar with, with 0.85 points). Then, efficiency (0.9), attractiveness (1.43) and dependability (1.3 points) are the next aspects in the list. The service’s novelty is very positively evaluated (1.75), together with stimulation (1.7).

7 Conclusions and further work

In this paper, we have explored different classification algorithms to build a real-time in-air hand shape identification system, ready to be integrated with non-critical smart space applications through low-cost devices such as Leap Motion. The study has compared the significance of intrinsic morphological hand features vs. pose hand features, showing that this second group is more relevant for the classification process. The use of ‘sweet spot’ feedback has been crucial to obtain a reasonable number of Correctly Classified Instances for all the tested algorithms. After ranking and giving weights (0–7) to the methods depending on their performance with respect to the CCI rate, time to build the model, response to training and scalability on the right hand dataset, the following ordered list is obtained: FT (22 points), LMT (18), Logistic (17), MP (16), Simple Logistic (15), IB1 (13), NNge (11) and SMO (8) (note that any criteria has been prioritized over any other). There are not sufficient reasons to defend a single algorithm as the best solution. Trees are offering the best global performances although could scale badly for an increasing number of users. Nearest Neighbor methods are very sensible to training, but once done, they perform reasonably well. Multilayer Perceptron is an option that becomes robust with respect to training and scales well.

On the results of the analysis of the algorithmic choices for classification, we have opted for using a nearest neighbor strategy to implement the real-time classifier. To provide an initial evaluation of the system performance and test how the users feel utilizing it in real settings, two different services have been implemented on top of the identification system. The first service enables the customization of the environment with the user’s preferred settings, while the second relies on identification to then facilitate gesture-based interaction with the objects around. We have carried out a preliminary user test with five users. Results of the test show that the obtained correct classification rate in real time is fairly lower than the off-line one. This is in part due to the difference between the hand pose in training and real time sessions. To achieve a correct classification, it is very important that the user replicates the training pose as close as possible. In practice, this is difficult, even with the sweet-spot feedback, at least until the user gets familiar to the device. On the user experience side, our test has shown that this space-embedded identification system is positively perceived by the users, in spite of its lack of efficiency.

The integration of the identification system with a gesture-based recognition system enabled by Leap shows the potential of merging both capabilities in a novel interaction flow. Leap-based gesture recognition for smart home interaction has demonstrated to be feasible; the subject-action-object grammar-based system obtains its lower rating on perspicuity factors, so simpler or more traditional approaches may be better accepted from the user side. Nevertheless, the approach can be suitable for specific user groups (e.g., impaired people). Further work thus needs to consider how to improve the accuracy of the classification and the sweet spot strategies, prior to extending the user experience study to a wider sample of users of different profiles.

Acknowledgments This work has been supported by the Spanish Ministry of Economy and Competitiveness under grant TEC2014-55146-R and by the Technical University of Madrid under grant RP150955017.

References

- Adán M, Adán A, Vázquez AS, Torres R (2008) Biometric verification/identification based on hands natural layout. *Image Vis Comput* 26(4):451–465
- Avola D, Petracca A, Placidi G, Spezialetti M, Cinque L, Levaldi S (2014) Markerless hand gesture interface based on LEAP motion controller. In: *Proc. of the international conference on distributed multimedia systems*
- Bergesio L, Marquinez I, Bernardos AM, Besada JA, Casar JR (2013) PERSEO: a system to personalize the environment response through smart phones and objects. In: *IEEE international conference on pervasive computing and communications workshops*, pp 640–645
- Bernardos AM, Sánchez JM, Portillo JI, Besada JA, Casar JR (2015) A contactless identification system based on hand shape features. *Procedia Comput Sci* 52:161–168
- de Santos-Sierra D, Arriaga-Gomez MF, Bailador G, Sanchez-Avila C (2014) Low computational cost multilayer graph-based segmentation algorithms for hand recognition on mobile phones. In: *International carnahan conference on security technology (ICCST)*, IEEE, pp 1–5
- Duta N (2009) A survey of biometric technology based on hand shape. *Pattern Recogn* 42(11):2797–2806
- Ferrer M, Fàbregas J, Faundez M, Alonso J B, Travieso C (2009) Hand geometry identification system performance. In: *43rd annual international carnahan conference on security technology*, IEEE, pp 167–171
- Guna J, Jakus G, Pogačnik M, Tomažič S, Sodnik J (2014) An analysis of the precision and reliability of the leap motion sensor and its suitability for static and dynamic tracking. *Sensors* 14(2):3702–3720
- Guo JM, Liu YF, Chu MH, Wu CC, Le TN (2011) Contact-free hand geometry identification system. In: *18th IEEE international conference image processing (ICIP)*, pp 3185–3188
- Han J, Gold N (2014) Lessons learned in exploring the Leap Motion TM sensor for gesture-based instrument design. In: *Proceedings of the international conference on new interfaces for musical expression*, pp 371–374
- Hassenzahl M, Tractinsky N (2006) User experience—a research agenda. *Behav Inf Technol* 25(2):91–917
- Jain AK, Duta N (1999) Deformable matching of hand shapes for user verification. In: *International conference on image processing*, IEEE, Vol 2, pp 857–861
- Jain AK, Flynn P, Ross AA (2007) *Handbook of biometrics*. Springer, Berlin
- Kanhangad V, Kumar A, Zhang D (2011) Contactless and pose invariant biometric identification using hand surface. *IEEE Trans Image Process* 20(5):1415–1424
- Kong A, Zhang D, Kamel M (2009) A survey of palmprint recognition. *Pattern Recogn* 42(7):1408–1418
- Kumar A, Zhang D (2006) Personal recognition using hand shape and texture. *IEEE Trans Image Process* 15(8):2454–2461
- Kumar A, Zhang D (2007) Hand-geometry recognition using entropy-based discretization. *IEEE Trans Inf Forensics Secur* 2(2):181–187
- Kumar A, Wong DC, Shen HC, Jain AK (2003) Personal verification using palmprint and hand geometry biometric. *Audio-and video-based biometric person authentication*. Springer, Berlin, pp 668–678
- Laugwitz B, Held T, Schrepp M (2008) *Construction and evaluation of a user experience questionnaire*. Springer, Berlin
- Marin G, Dominio F, Zanuttigh P (2015) Hand gesture recognition with jointly calibrated Leap Motion and depth sensor. *Multimed Tools Appl* 1–25. doi:10.1007/s11042-015-2451-6
- Miller RP (1971) Finger dimension comparison identification system. US Patent No. 3576538
- Morales A, Ferrer MA, Díaz F, Alonso JB, Travieso CM (2008) Contact-free hand biometric system for real environments. In: *16th European signal processing conference, IEEE*, pp 1–5
- Mostayed A, Kabir ME, Khan SZ, Mazumder MG (2009) Biometric authentication from low resolution hand images using radon transform. In: *12th international conference on computers and information technology, IEEE*, pp 587–592
- Nielsen J, Landauer TK (1993) A mathematical model of the finding of usability problems. In: *Proc. of ACM INTERCHI'93 conference, Amsterdam, The Netherlands*, pp 206–213
- Oden C, Ercil A, Buke B (2003) Combining implicit polynomials and geometric features for hand recognition. *Pattern Recogn Lett* 24(13):2145–2152
- Pavešić N, Ribarić S, Ribarić D (2004) Personal authentication using hand-geometry and palmprint features—the state of the art. *Hand* 11:12
- Ross A, Jain AK (1999) A prototype hand geometry-based verification system. In: *Proceedings of 2nd conference on audio and video based biometric person authentication*, pp 166–171
- Sanchez-Reillo R, Sanchez-Avila C, Gonzalez-Marcos A (2000) Biometric identification through hand geometry measurements. *IEEE Trans Pattern Anal Mach Intell* 22(10):1168–1171
- Sodhi GS, Kaur J (2003) Indian civilization and the science of fingerprinting. *Indian J Tradit Knowl* 2(2):126–136
- Tistarelli M, Schouten B (2011) Biometrics in ambient intelligence. *J Ambient Intell Humaniz Comput* 2(2):113–126
- Wang X, Bernardos AM, Besada JA, Metola E, Casar JR (2015) A gesture-based method for natural interaction in smart spaces. *J Ambient Intell Smart Environ* 7(4):535–562
- Wong AL, Shi P (2002) Peg-free hand geometry recognition using hierarchical geometry and shape matching. *IAPR workshop on machine vision applications*, pp 281–284
- Woodard DL, Flynn PJ (2005) Finger surface as a biometric identifier. *Comput Vis Image Underst* 100(3):357–384
- Yörük E, Dutağacı H, Sankur B (2006) Hand biometrics. *Image Vis Comput* 24(5):483–497
- Yuan Y, Barner K (2006) An active shape model based tactile hand shape recognition with support vector machines. In: *40th annual conference on information sciences and systems, IEEE*, pp 1611–1616
- Zheng G, Wang CJ, Boulte TE (2007) Application of projective invariants in hand geometry biometrics. *IEEE Trans Inf Forensics Secur* 2(4):758–768