Univ Access Inf Soc (2014) 13:257–275
DOI 10.1007/s10209-013-0317-0

LONG PAPER

Using hand gestures to control mobile spoken dialogue systems

Nikos Tsourakis

Published online: 5 October 2013

Abstract Speech and hand gestures offer the most natural modalities for everyday human-to-human interaction. The availability of diverse spoken dialogue applications and the proliferation of accelerometers on consumer electronics allow the introduction of new interaction paradigms based on speech and gestures. Little attention has been paid, however, to the manipulation of spoken dialogue systems (SDS) through gestures. Situation-induced disabilities or real disabilities are determinant factors that motivate this type of interaction. In this paper, six concise and intuitively meaningful gestures are proposed that can be used to trigger the commands in any SDS. Using different machine learning techniques, a classification error for the gesture patterns of less than 5 % is achieved, and the proposed set of gestures is compared to ones proposed by users. Examining the social acceptability of the specific interaction scheme, high levels of acceptance for public use are encountered. An experiment was conducted comparing a button-enabled and a gesture-enabled interface, which showed that the latter imposes little additional mental and physical effort. Finally, results are provided after recruiting a male subject with spastic cerebral palsy, a blind female user, and an elderly female person.

Keywords Gestured-controlled mobile applications · Gesture and speech interfaces · Gesture classification · Mobile accessibility

N. Tsourakis (✉)
ISSCO/TIM/FTI, University of Geneva, Geneva, Switzerland
e-mail: Nikolaos.Tsourakis@unige.ch

1 Introduction and motivation

According to [12], people prefer a combination of speech and gestures over speech and gestures alone while interacting with a computer system. The proliferation of mobile devices imposes new patterns of interaction as these devices usually compete for the same human resources needed for other mobility tasks [17] and as users, while mobile, perceive information differently [27]. Although previous work provides some guidelines regarding gesture-based interfaces [14, 24], little attention has been paid to the question of how to control spoken dialogue systems (SDS) with gestures, while most efforts were directed to parallel combine these two distinct input modalities in order to control multimodal interfaces [20, 21]. A notable exception is the newly introduced feature of iPhone's Siri, which activates the microphone after lifting the handset to the ear.

This work tries to alleviate this deficiency by introducing a novel solution to the problem, where concise and intuitively meaningful gestures are used to trigger the commands to any SDS. Specifically, a set of six gestures is used for moving forward and backward in the dialogue flow, starting and stopping speaking, getting help, and aborting an ongoing action. As a proof of concept, these gestures have been incorporated in the mobile version of the CALL-SLT system [3], which is a spoken conversational partner designed for beginner- to intermediate-level language students who wish to improve their spoken fluency in a limited domain.

Special kinds of disabilities related to user's current situation can pose hurdles to the efficient usage of a mobile speech system. Anyone who has tried using a similar application with one hand while carrying a child, reading the screen display during a sunny day, or interacting with

the screen while wearing gloves knows how he or she can become "effectively" impaired. The concept of "situation-induced disabilities" [37] has been introduced to describe similar non-optimal conditions where the user's behavior is dictated by both the environmental conditions and the characteristics of the device. Although the move in the direction of gesture-driven interfaces was motivated by feedback from non-disabled people who have used the application, it becomes apparent that all the arguments apply even more strongly to users who are vision-impaired or lack fine motor control. The coordination required to use the normal button-controlled interface is experienced as challenging by many non-disabled people and would be beyond the reach of almost all users who experience problems with sight or fine motor skills.

In contrast, it is likely that the gesture-based interface could be operated in many of these situations. If, for example, the device is strapped to the user's hand (like a smart watch [26]), it can be operated using only gross motor movements. The fact that gesture identification is trained from the user's own repertoire of movements means that it can potentially be adapted to a wide range of conditions.

In this work, apart from introducing the gestures, eight users were asked to perform and to evaluate them. Using machine learning techniques, the aim was to quantify how well each gesture pattern can be separated and thus obtain a good estimate of what can be expected from a future deployed system. Participants were also asked to propose their own set of gestures and evaluate the ones presented by us. The social acceptability of this type of interaction was also examined, since handheld devices are part of one's public appearance. Finally, eight participants were asked to use CALL-SLT using both the button-enabled and gesture-enabled interfaces. Tests were also performed with a male subject with mild cerebral palsy, a blind female user, and an elderly female person.

The rest of the paper is organized as follows. Section 2 describes the CALL-SLT gesture-based interface, and Sect. 3 describes the data collection protocol. Section 4 presents a series of experiments designed to evaluate performance issues. The final section concludes.

2 Gesture-driven interfaces

Gesture-driven interfaces augment traditional graphical user interfaces by incorporating specific hand poses, spatial trajectories of the hands or stylus, motions to indicate an object, or motions of almost any body part [25]. The growing interest in multimodal interface design is inspired largely by the need to offer friendlier interfaces that allow a more natural user interaction. Gestures are an alternative or complementary modality for application control. There is a broad spectrum of hardware and software applications that leverage gestures as an input source especially in the game industry (cf. Microsoft Kinect, Nitendo Wiimote) as well as hundreds of mobile accelerometer-based applications for Android and iOS.

Different technologies can be used to capture these gestures either in *active* or in *passive* mode. Dedicated devices such as position trackers or sensing data gloves can be incorporated in the active mode [18]. In passive mode, user input can be monitored with one or more cameras, and computer vision algorithms are used to segment and classify the image data [4]. While passive modes may be "attentive" and less obtrusive, active modes generally are more reliable indicators of user intent [29]. The interface that will be described in the next subsection works in active mode.

In everyday life, people may use gestures as the only means of communication; in most cases, however, gestures occur along with other modalities such as speech. Since the appearance of the "Put-That-There" demonstration system [2], which processed speech in parallel with touch-pad pointing, a variety of new multimodal systems that utilize hand gestures have emerged. Most efforts have been directed toward seamlessly combining speech and gestures in order to control multimodal interfaces [20, 21], while others have focused on the synergies among them to accomplish a task [31, 39]. Gestures have also been incorporated in physical spaces for interacting with large displays [28] or with digital home environments [40]. Additionally, work from [8] investigated the usability of gestures and how they could be used to express the most frequently used remote control commands. Studies agree, however, that different people usually prefer different gestures for the same task [16, 28].

2.1 A gesture-based interface

CALL-SLT is a generic multilingual Open Source platform based on the "spoken translation game" idea of [42]. The core idea is to give the student a prompt, formulated in their own (L1) language, indicating what they are supposed to say; the student then speaks in the learning (L2) language and is scored on the quality of their response. When the student has practiced sufficiently on the current prompt, they can ask for the next one. At any time, they can request help; the system responds by giving textual and/or spoken representations of a correct response to the current prompt. A detailed overview of CALL-SLT functionality can be found in [3] and the top-level software architecture of the system in [9].

The system also offers several ways to control both the flow of prompts and the way in which the matching process

is performed. For example, prompts are grouped into lessons, each of which will typically be arranged around a theme, and recognition can be adjusted so as to make it more or less forgiving of imperfect pronunciation. The student will sometimes use these features, perhaps selecting a new lesson or making the recognition more forgiving if they are having difficulties. Most of the time, however, they will be in an interaction loop which only uses a small set of core commands. They get the next prompt, optionally ask for help, start recognition, stop it when they have finished speaking, and see whether the system accepted their spoken response. If it did, they move to the next prompt; otherwise, they try again. It is consequently very important to make the core commands ergonomically efficient. The left side of Fig. 1 shows a screenshot of the GUI for the mobile version of the CALL-SLT system, whereas in the right side some typical readings of the accelerometer are presented.

For the mobile version of the system, a button- controlled interface poses many problems. Few users will have a headset, and the majority will use the tablet's onboard microphone; this involves lifting the tablet to the user's mouth while speaking and makes a push-and-hold interface extremely inconvenient.

Another important point is that there is no tactile feedback from the touch screen, increasing the user's uncertainty about the interaction status. All of these problems become more acute when one considers that a crucial point of deployment on a mobile device is to be able to access the system in outdoor environments, where the screen is less easily visible and the user may be walking or inside a moving vehicle.

For these reasons, the use of an interface has been investigated, which controls the key CALL-SLT functionalities using the intuitive gestures shown on Fig. 2. The current version of the interface supports six gestures. "Get next prompt" and "Return to previous prompt" are signaled by tipping the tablet right and left. "Start recognition" is triggered by moving the tablet so that the microphone is in front of the user's mouth (this involves rotating the device by about 90°, since the Galaxy Tab's microphone is on the upper left side), and "End recognition" is triggered by moving the tablet away from the mouth again. "Help" is requested by moving the device so that the speaker is next to the subject's ear, the natural position for listening to spoken help in a noisy environment. "Abort" is signaled by shaking the device from side to side. In essence, these gestures constitute the minimum set that covers the basic functionalities of any spoken dialogue system.

3 Data collection

Galaxy Tab's onboard accelerometer was used, which returns measurements of the G-force experienced by the device along each of the three component axes, and sampled these values every 50 ms for one second while performing examples of the six commands. Twenty examples of each command from eight subjects were collected, half male and half female, between 20 and 50 years old with higher academic education; half of them had no IT background. The six right-handed subjects used the device as depicted in the diagram (Fig. 2), holding it in their left hand while seated. The registration of each gesture was initiated by pressing a start button. This has the advantage that each interaction starts from the initial position and that the acquired accelerometer data correspond only to the gesture performed.

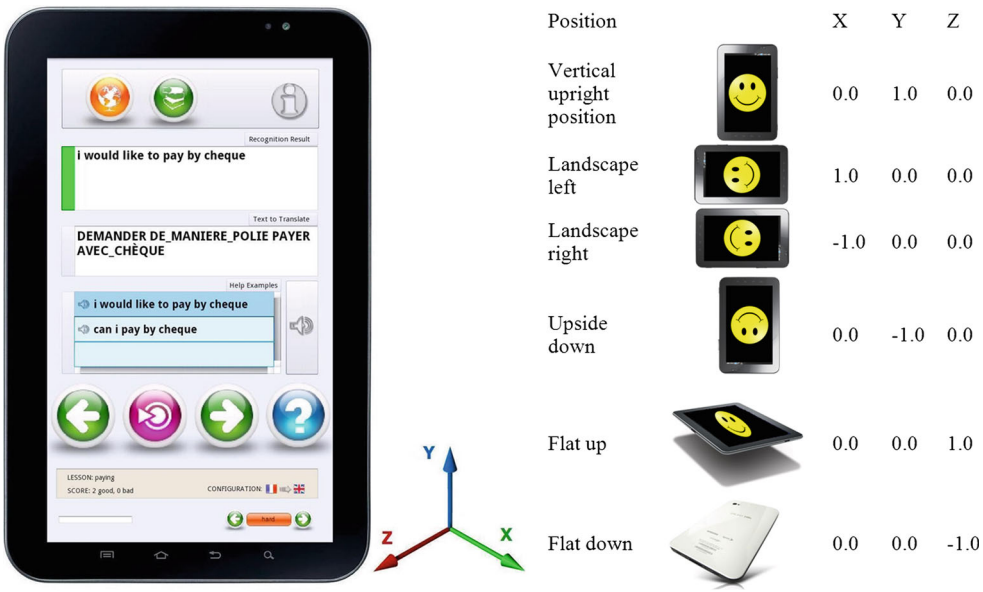

Fig. 1 *Left* CALL-SLT English-for-French application running on the Samsung Galaxy Tab. The *middle pane* shows the prompt; the *top pane*, the recognition result; the *bottom pane*, text help examples. Button controls are arranged along the bottom. *Right* Typical readings of the axes when the device is in various positions

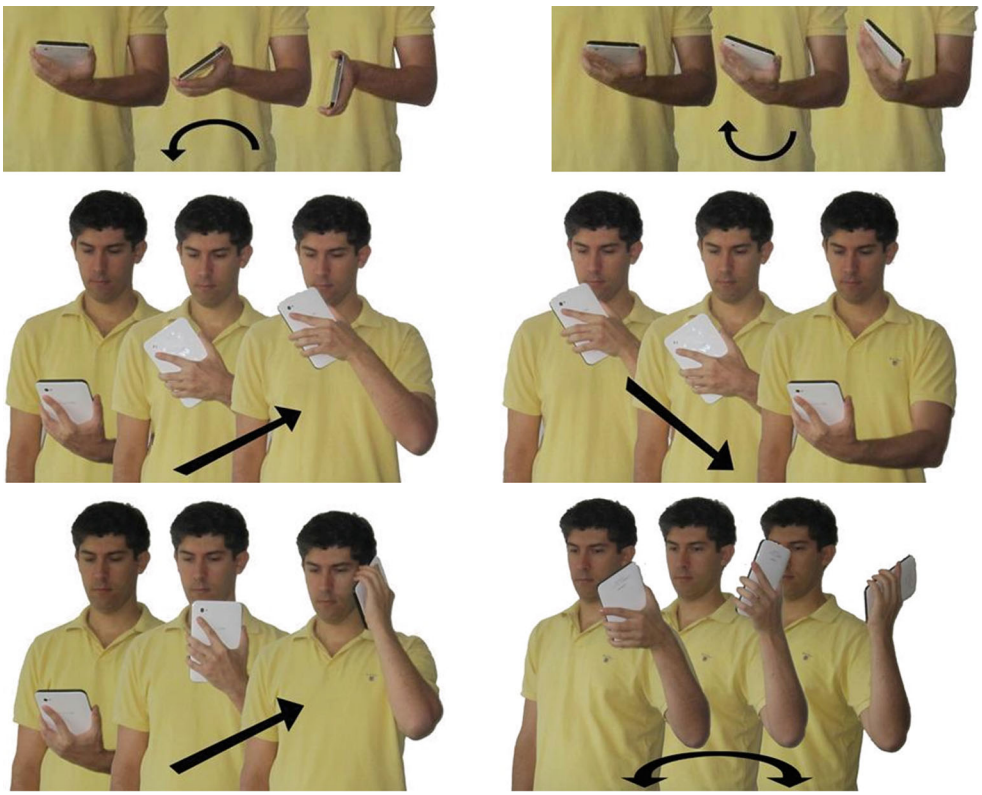

Fig. 2 Proposed gestures set: From left to right, bottom down *next*, *previous*, *start recognize*, *stop recognize*, *help*, *abort*

This configuration is the natural one for a right-handed person; they hold the tablet in their left hand, since they wish to press the buttons with the fingers of their right hand. The two left-handed subjects held the device in their right hand and used their left hand to manipulate the controls. Similar data were also collected for eight common non-gesture conditions shown in Table 1.

The mean and Root Mean Square (RMS) values for the *X*, *Y* and *Z* axis components were extracted and used as the main features. RMS is a useful statistical measure when variates are positive and negative as in this case. The plots in Fig. 3 show the data-points for the *XY* plane, tagged by gesture, for one of the subjects. Even with a very basic feature-space, Fig. 3 suggests that the gestures should be easy to separate from each other.

4 Experiments

4.1 Gestures classification

Different methodologies have been proposed in the literature for performing the classification of gestures, e.g. Dynamic Bayesian Networks [5], Support Vector Machines (SVM) [32, 44], Hidden Markov Models (HMM) [15] and Dynamic Time Warping and k-means clustering [6]. The trade-off among these methods is in accuracy versus the processing time and the amount of training data required. The following three subsections present an experimentation with a subset of these methodologies by utilizing either features from the acceleration vectors or all the data available.

Table 1 Non-gesture movements used in experiment

Lying	The device is lying on the table
Sitting, holding	The user is sitting, holding the device in front of him
Standing, holding	The user is standing, holding the device in front of him
Standing, relaxing	The user is standing, holding the device vertically
Running	The user is running
Climbing	The user is climbing a flight of stairs
Descending	The user is descending a flight of stairs
Walking	The user is walking

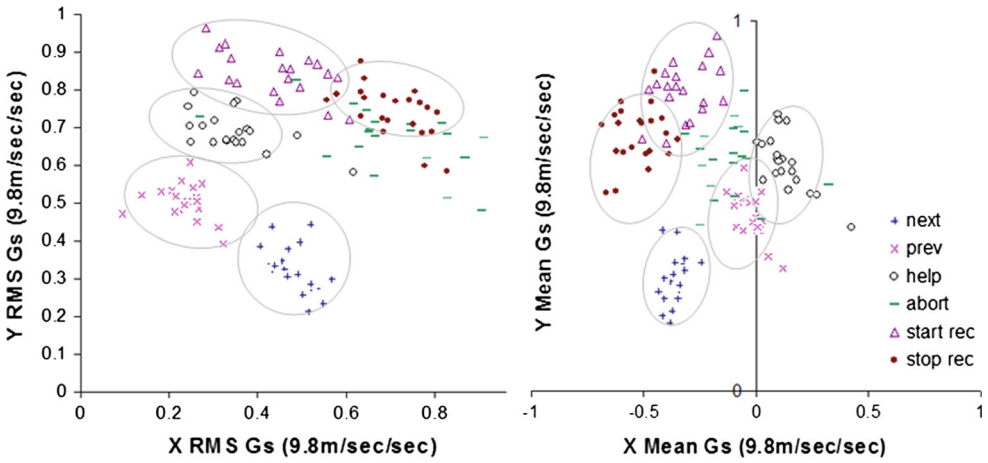

Fig. 3 Separation of gestures in acceleration-space: RMS (*left*) and mean (*right*) values of the *X* and *Y* components of acceleration for one of the subjects

4.1.1 Feature-based classification

Experimentation with some standard machine learning algorithms confirmed the intuitive impression that the gestures could easily be separated, and also showed that the gestures could be separated reasonably well from the non-gesture conditions. For each subject, 75 % of the data (both gesture and non-gesture) were used for training and 25 % for testing. Classification was performed using Naive Bayes, Ensembles of Nested Dichotomies [7], Multilayer Perceptron with back propagation (one hidden layer with 10 hidden nodes, learning rate 0.3 and momentum 0.2, 500 epochs sigmoid for activation), Decision Trees implementing C4.5 pruned algorithm, Random Forest of 10 trees considering 4 random features classifiers and Functional Trees [10], SVM (polynomial kernel and trade-off between training error and margin 5,000), and Nearest-neighbor using non-nested generalized exemplars [23].

The results of the different classification methods using the Weka Toolkit [11] are shown in Table 2, where it can be seen that most of the methods offer low error rates. Note that the classification tasks were performed using data from both gesture and non-gesture movements. The confusion matrix for SVMs presented in Table 3 provides a better overview of the classification task. As one can observe, the "descending"' movement seems to cause the most recognition errors, where only 24 out of 40 test samples (60 %) were correctly classified. In general the six gestures of interest can be easily recognized.

Table 2 Classification error (percentage) on gesture recognition using 8 classifiers

Classifier	6 Features(*X*-Mean, *Y*-Mean, *Z*-Mean, *X*-RMS, *Y*-RMS, *Z*-RMS)			
	Correctly classified (%)	Precision (%)	Recall (%)	F-measure (%)
Naïve bayes	91.61	92.48	91.61	91.64
END	90.18	91.14	90.20	89.71
SVM	92.50	92.81	92.50	92.34
Decision tree C4.5	87.14	88.45	87.15	86.45
Functional trees	90.89	91.75	90.90	90.81
Random forest	89.82	90.44	89.84	89.4
Nearest neighbor	93.39	94.45	93.41	93.01
Multilayer Perceptron	92.50	93.19	92.51	92.29

4.1.2 Hidden Markov Model classification

The analysis in the previous subsection was based on features extracted from the sampled acceleration frames (*X*, *Y*, *Z* values every 50 ms). In this subsection and the following, two different classification methods that process each one of the frames instead of the calculated features will be applied. The immediate benefit of feature extraction is the dimensionality reduction, which can offer faster processing times and reduced storage sizes. However, when these issues are not of prime importance the exploitation of every single data element by statistical models like HMM can offer better results.

HMMs have been extensively used in speech recognition systems, and due to their ability to classify temporal data of no fixed length, they are a good candidate for gesture recognition. Different studies claim high gesture recognition rates; according to [16] up to 98.8 %, according to [36] between 85 and 95 %, and according to [33] 97.6 % on average. The results shown in Table 4 were produced after training a left-to-right HMM with six states in the Weka Toolkit, for each gesture and user.

Continuing the analysis, the aim was to investigate the effects of vector quantization on the data. As it has been already mentioned the accelerometer was sampled every

Table 3 Confusion matrix for the support vector machine classifier

	Movements	a	b	c	d	e	f	g	h	i	j	k	l	m	n
a	Next	38	0	0	0	2	0	0	0	0	0	0	0	0	0
b	Previous	0	37	2	1	0	0	0	0	0	0	0	0	0	0
c	Help	0	3	36	1	0	0	0	0	0	0	0	0	0	0
d	Abort	0	0	1	39	0	0	0	0	0	0	0	0	0	0
e	Start recognition	0	0	0	0	38	2	0	0	0	0	0	0	0	0
f	Stop recognition	1	0	0	0	3	34	0	0	1	0	0	0	1	0
g	Lying	0	0	0	0	0	0	40	0	0	0	0	0	0	0
h	Sitting, holding	0	0	0	0	0	0	0	40	0	0	0	0	0	0
i	Standing, holding	0	0	0	0	0	0	0	0	40	0	0	0	0	0
j	Standing, relaxing	0	0	0	0	0	0	0	0	0	40	0	0	0	0
k	Running	0	0	0	0	0	0	0	0	0	0	40	0	0	0
l	Climbing	0	0	0	0	0	0	0	0	0	0	0	32	8	0
m	Descending	6	0	0	0	0	1	0	0	0	0	0	9	24	0
n	Walking	0	0	0	0	0	0	0	0	0	0	0	0	0	40

Table 4 Classification error (percentage) on gesture recognition using HMM

Classifier	Use the *X*, *Y*, *Z* acceleration frames (sampled every 50 ms for 1 s)			
	Correctly classified (%)	Precision (%)	Recall (%)	F-measure (%)
HMM	95.54	96.36	95.53	95.34

50 ms for 1 s yielding a sequence of frames containing the *X*, *Y*, *Z* acceleration force. As depicted in Fig. 4, the quantization process of the training data is the following:

1. Get the input movement (next, previous, walking, etc).
2. Eliminate similar frames using the Euclidean distance. Keep the frames that are dissimilar above a cutoff threshold. This was empirically chosen equal to 0.055.
3. If *n* is the desired codebook size and *m* the frame array size, cluster the frames into *n* groups. Besides hierarchical and k-means clustering one can create a codebook with *n* random vectors from the frame array (**random**) or sort the frame array and get the vectors at position *m*/*n*, 2 *m*/*n*, …, *nm*/*n*(**simple**).
4. The result is a codebook for each movement, clustering method and codebook size.

Table 5 summarizes the percentages of input vector that remained for the follow-up analysis after performing the preprocessing step. The results provide an indirect indication of how complex a gesture is. For example, if you just sit and hold the device in front of you, the remaining vectors are 6.46 % of the initial ones, whereas if you descend a flight of stairs the ratio rises to its highest value of 92.32 %. A correlation test was performed between the gesture recognition error using the SVM classifier and the remained vector percentage; it was found that the two variables are negative correlated (Pearson's $r(12) = -0.54$, $p < 0.05$), so the gesture complexity has an impact on the recognition performance.

Figure 5 presents a visualization of the "next" gesture acceleration vectors after the clustering process. As it can be observed, the methods offer a quite good distribution of prototype vectors of the sample vectors. During the testing phase the 3-dimensional vectors which are less distant than 0.055 from the preceding vector are filtered out. The vector quantizer maps the remaining input vectors to codebooks of sizes 8, 14, 20 or 28. All movement codebooks with the same size were merged into a single one and the HMM classification produced the results presented in Fig. 6. The hierarchical clustering seems to outperform the others; when using codebooks with more than 14 vectors the results are comparable to the ones of Sect. 4.1.1.

Two-way ANOVA, identified significant main effects of clustering method ($F(3,127) = 7.32$, $p < 0.001$) and codebook size ($F(3,127) = 16.67$, $p < 0.001$) on the correct classification rate. A post-hoc Tukey's HSD ($p < 0.05$) pairwise comparison revealed the significant differences shown in Table 6.

4.1.3 Template classification

Unlike the machine learning and statistical methods presented in the previous subsections that require sufficient number of samples to be trained, it is often desirable to use alternative classification methods based on template matching. These can start working even with one sample per gesture and thus minimize training time. In this subsection, the $1 recognizer [43] has been incorporated,

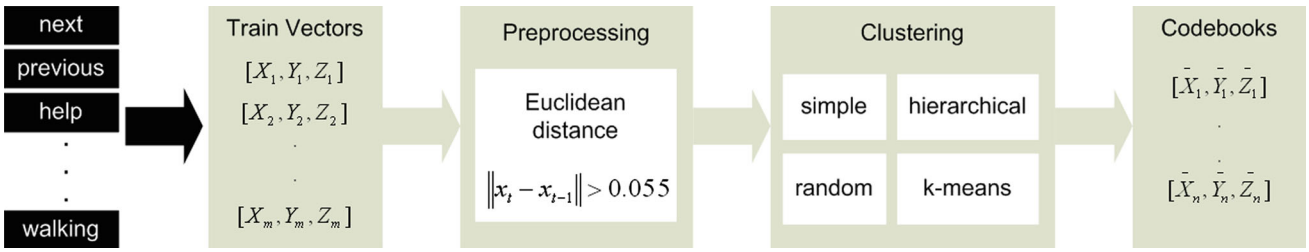

Fig. 4 Quantization process using the training data of each movement

Table 5 Rate of the input vectors that remained after the preprocessing step

	next	previous	help	abort	start	stop	lying
Rate	77.41%	84.11%	81.84%	87.82%	79.79%	78.34%	6.3%
	sitting holding	standing holding	standing relaxing	running	climbing	descending	walking
Rate	6.46%	7.35%	29.59%	88.25%	84.64%	92.32%	85.56%

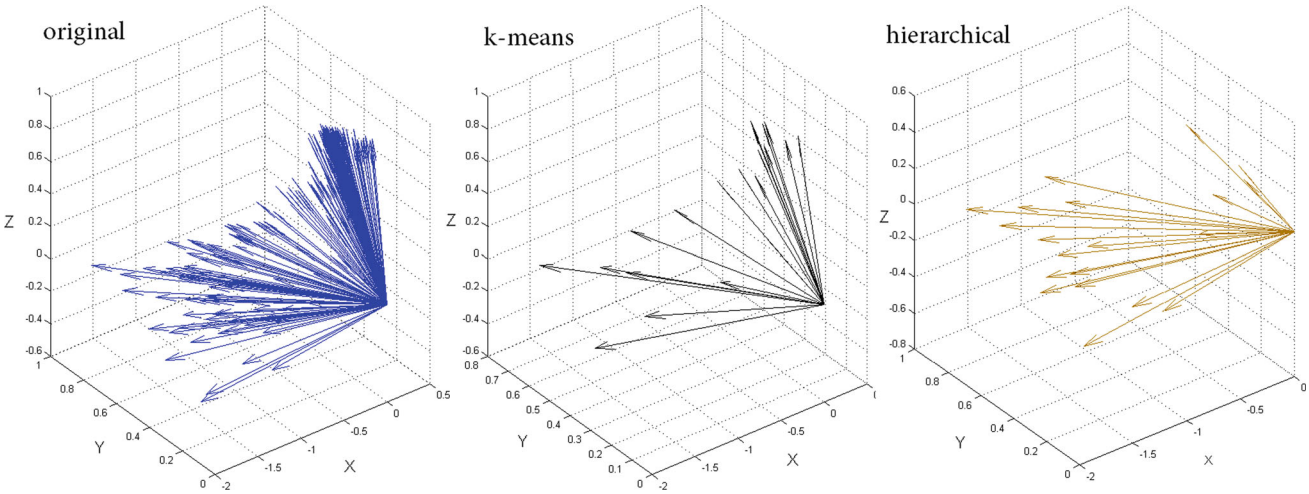

Fig. 5 Quantization of the "next" gesture using different clustering methods (codebook size = 20)

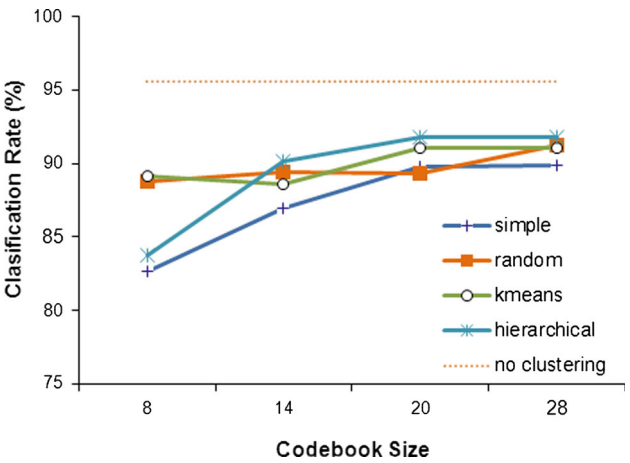

Fig. 6 Classification with HMMs using different clustering methods and codebook sizes

which is a small footprint recognizer of gestures made by path-making instruments like pens and fingers in the two dimensional space; according to the authors, it can achieve 99 % accuracy of recognizing 2-D single-stroke paths on a display. The $1 recognizer performs template classification by matching the geometric specifications of two hand-writings. The algorithm involves four steps: (1) resample the input points, (2) rotate the points at 0°, (3) scale points in a bounding box and (4) match points against a set of templates. Despite the fact that the gestures in this case study are performed in the three-dimensional space, it was desired to investigate the classification performance of this approach in the *XY*, *XZ* and *YZ* planes.

In a similar manner as before, 75 % of the data (both gesture and non-gesture) were used for training and 25 % for testing. Figure 7(left) shows the results of the

Table 6 Significance difference of clustering methods and codebook sizes in pairwise comparisons using Tukey's HSD test

	simple	random	k-means	hierarchical
simple		<0.001	<0.001	<0.01
random			0.99	0.99
k-means				0.98
hierarchical				
	8	14	20	28
8		<0.001	<0.001	<0.001
14			0.18	0.14
20				1
28				

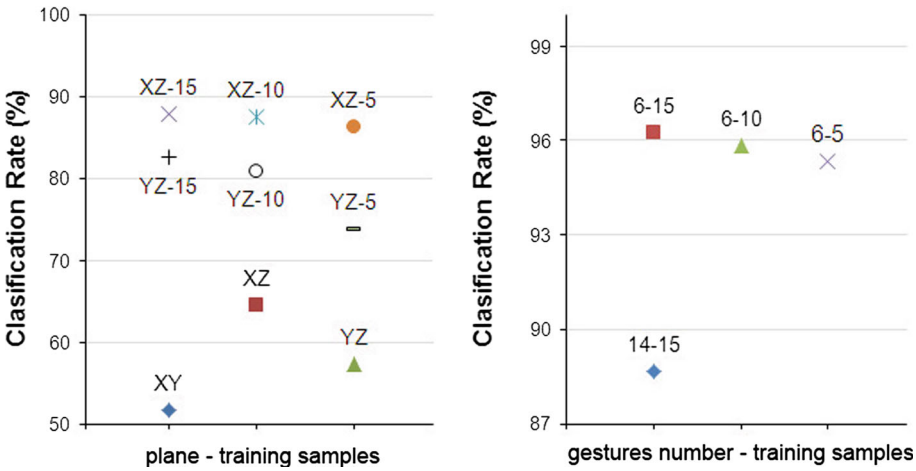

Fig. 7 *Left* Classification using the $1 recognizer in the *XY*, *XZ*, *YZ* planes and with different size of training data. *Right* Classification using the uWave algorithm with different size of training data

recognition performance (*XY*, *XZ*, *YZ*), where the *XZ* plane demonstrates the highest correct classification rate. For real applications, however, this is far from acceptable. Therefore, the analysis was repeated by removing the non-gesture movements (sitting, walking, etc) as a compromise with what a user might do during interaction with the system. Moreover, the number of training samples (15, 10, 5) was altered, and the corresponding results are also depicted in Fig. 7(left) for the *XZ* and *YZ* planes. The best rate is again for *XZ* with 15 training samples and it is equal to 87.92 %. One-tail *t*-tests between pairs of *XZ*, *YZ* for the same number of training samples show statistical significant differences. Specifically, for 15 samples: $t = 2.31$, $df = 7$, $p < 0.01$, for 10 samples: $t = 2.94$, $df = 7$, $p < 0.01$ and for 5 samples: $t = 3.52$, $df = 7$, $p < 0.005$.

Although the results are less than optimal, the developers may benefit from the low requirements of this approach by using an even smaller set of gestures or introducing an alternative, easier recognizable set. However, a more promising approach is to combine the recognition results in the different planes and ultimately to implement a similar algorithm in three-dimensional space.

To finesse the limitations of the previous template matching technique, which can be used efficiently for certain types of user interface gestures, the uWave algorithm [22] was incorporated based on Dynamic Time Warping (DTW) in order to classify our gestures. The data are used again directly without doing any feature extraction and are processed in the time domain as specified by the DTW. The algorithm bases recognition on the matching of two time series of forces, measured by the single three-axis accelerometer. The analysis yields to a recognition accuracy result equal to 88.66 % (Fig. 7(right)). As before, the minimum set of gestures (6 gestures) and different sizes of training data sets (15, 10, 5) were used. When all the training data are used the correct classification rate is 96.25 %. A single-factor ANOVA showed no significant differences in the classifications rates for the various sizes of the training data.

4.2 Gestures survey

Before providing the data analyzed in the previous subsection, the same users were asked to participate in an evaluation of the proposed gesture set. After a short introduction of the non-gesture GUI and the presentation of a short video clip, they had to improvise gestures that would provide the same functionality. It was emphasized to the users that help is acoustic as well as visual and that one had to speak close to the microphone of the device. Following the presentation of the gesture repertoire, the users were asked to fill out a questionnaire that asked how difficult it was to perform each gesture, if it was intuitive or not, and if they preferred it to their own

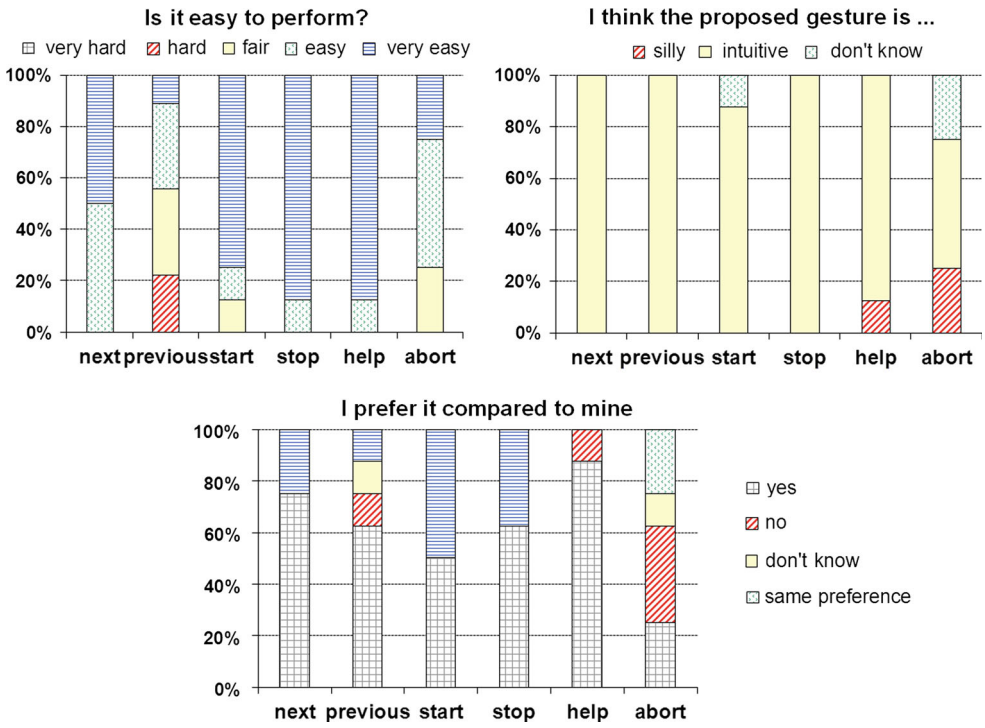

Fig. 8 Charts of the easiness, impression and preference for each one of the proposed gestures

suggestion. The results of this survey are shown in Fig. 8.

As one can observe, most of the subjects agree that the proposed gestures are easy to perform and are intuitive. They also prefer the proposed set compared to theirs, with a small exception on the "abort" gesture. It is suggested that this has to do with the user's personal feelings concerning the specific movement. As a matter of fact three of them had chosen the same gesture for "abort"; just flip the device, related to the metaphor of how you hang up the telephone set. According to another user, this metaphor applies when one is using the system inside the car; the user simply puts the device down to signify "stop recognizing". Cultural differences were also encountered as one subject proposed for "help" the hand gesture that signifies question for many Greeks (rotating clockwise the palm close to the face). Apart from one subject, all participants recommended gestures that were easy to execute. Finally, one of the participants suggested that he would prefer an interface that combined both hand gestures and voice commands.

It is not proposed that this suggestion on how to perform each gesture is unique and applicable to any person. As stated in the introduction of this paper, the idea is to train the system from the user's own repertoire of movements, which can obviously change between user types and conditions. In another domain (interacting with large displays) different subjects seemed to prefer different gestures for the same activity [28], something that was expected to encounter in the present case. Moreover, the tablet used has a physical size significantly larger than that of a typical smartphone, so one may reasonably argue that the proposed gesture set is not applicable to all devices. From the authors' point of view there is a lack of a large scale metaphor for gesture-based mobile SDS. Visual user interfaces have significantly benefited from the introduction of WIMP widgets that offer a unified interaction scheme. A new WIMP-based interface can rely on the knowledge accumulated over the years so that users do not need to learn new ways of doing things. However, a good analogy for gesture-based interfaces is lacking, so the work presented in the current paper can be considered as a contribution toward this direction.

4.3 Social acceptability

As well as trying to determine how well gesture recognition works or if users prefer the proposed set of gestures to theirs, another follow-up question was whether users would be willing to execute them in public. Although much work has been carried out on the technical aspects of gesture recognition, little attention has been paid to the social acceptability of interacting using gestures. Notable exceptions are [34, 35]. Social factors have an influence on technology acceptance [19], so it is necessary to offer guidelines for the design and evaluation of socially acceptable gestures. Therefore, the study continued by asking the same subjects as before to identify in which location (6 alternatives) and in front of which audience (6 alternatives) they would be willing to execute each of the

Table 7 Location and audience checklist

In which locations would you use this gesture? (check all that apply):	Who would you perform this gesture in front of? (check all that apply):
☐ Home	☐ Alone
☐ Pavement or Sidewalk	☐ Partner
☐ While Driving	☐ Friends
☐ As a Passenger on a Bus or Train	☐ Colleagues
☐ Pub or Restaurant	☐ Strangers
☐ Workplace	☐ Family

proposed gestures. As our focus has been on the gesture modality, we made clear to subjects that their answers should be irrelevant to the type of the application used (in our case a language learning system). The corresponding checklist is shown in Table 7.

The plots of Fig. 9 were constructed according to the users' answers. As it can be observed, the proposed set of gestures receives a high level of acceptability even in public places. Pavements, public transportation and workplaces do not impose any usage limitations. On the other hand, users seem reluctant to interact using gestures while driving, probably due to safety reasons as explicitly reported by many of them. Concerning the audience of usage, there is a universal positive agreement with a small exception on the "abort" gesture, which, as seen in the previous subjection, was the most controversial one. Compared to the aforementioned studies, the intuitiveness of the proposed gestures for the specific applications task has a beneficial impact on their social acceptance. During the design phase, effort was made to design the gestures as simple as possible and also to exploit any commonly acceptable interaction pattern. By putting the device close to the ear (help) or in front of the mouth (start recognition), a user simply re-uses patterns that have long been available. Likewise, the execution of "next" and "previous" commands resemble to playing a mobile video game. Conversely, executing "abort" in public areas may attract undesired attention.

In order to statistically verify the differences presented in Fig. 9(down), a significance test was performed. The response variables of Table 7 can take two possible outcomes (coded as 0 and 1), so a Cochran's Q test was executed. It was found that there exist significant differences in gesture usage in diverse places ($X^2(5) = 106.9$,

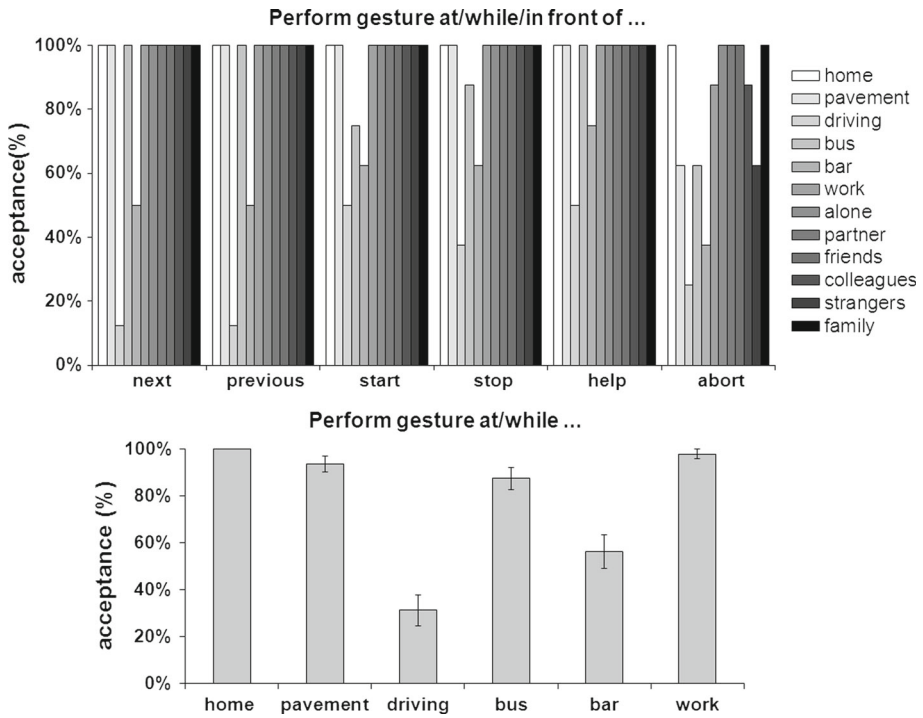

Fig. 9 Average percentage of gestures acceptability in different locations and in front of different people (*error bars* show one standard deviation)

Table 8 Significance difference of places in pairwise comparisons using continuity-corrected McNemar's tests with Bonferroni correction

	Home	Pavement	Driving	Bus/Train	Bar/Restaurant	Work
Home		1	<0.001	0.653	<0.001	1
Pavement			<0.001	0.992	<0.001	1
Driving				<0.001	0.017	<0.001
Bus/Train					<0.001	0.147
Bar/Restaurant						<0.001
Work						

$p < 0.001$). A pairwise comparison using continuity-corrected McNemar's tests with Bonferroni correction revealed what the significant differences are, as shown in Table 8.

4.4 Interacting with gestures

In the next part of this work a user study was conducted, where subjects were asked to use both the button-enabled and the gesture-enabled versions of the mobile CALL-SLT system. Specifically, 8 right-handed participants between 20 and 40 years old were recruited and asked to use the proposed set of gestures. It was decided to use native L2 speakers (3 French, 3 Greeks, and 2 Germans) to avoid excess recognition errors that could skew the aim of the study. Each experiment was completed when 30 spoken interactions were performed. Users had to follow a specific pattern which included three steps; going back or forward in the prompt list, asking for help and initiating recognition. It was ensured that the list of prompts contained around 20 elements so that subjects would practice both "next" and "previous" gestures. Participants started either with the button version or with one of the gesture-enabled versions while sitting in an office environment. They also used the application after it was trained with their own personal data. In the gesture-based interface shown in Fig. 10(left), the button bar has been replaced with an image.

Due to source code availability and implementation easiness it was decided to transcribe the SVM classification algorithm of [13] in Actionscript 3.0. The specific implementation concerned only the recognition part, whereas the training task using participants' data was done offline. For the specific test only 8 of the movements presented earlier were included (6 gestures + sitting holding + lying). On average the recognition algorithm running on the device takes 7.6 ms (SD = 2.7 ms). The initial design of the experiment presupposed that the accelerometer would always be on. However, an initial pilot study revealed the deficiencies of this approach as the gesture recognition error was too high for any real experimentation. Although one might argue that a different classification method could offer better results, this is not the case. As shown in Sect. 4.1, most of the errors originate from the non-gesture movements, which even after being removed from the training corpus did not yield any significant improvement

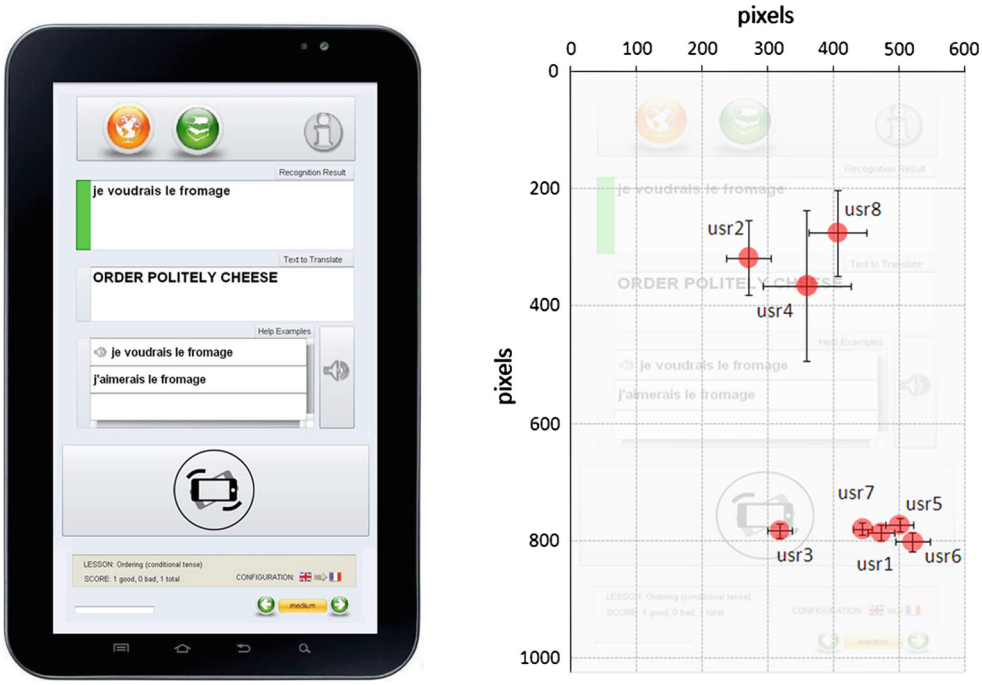

Fig. 10 *Left* Gesture-based interface. *Right* Scatter plot of the screen points chosen by users to initiate gesture recognition. Users with odd id started with the buttons version (*error bars* show one standard deviation)

during user tests. Essentially, the main problem is that the system does not know when the gesture starts. Therefore, polling the accelerometer every 50 ms for one second might not give the whole data range of the gesture the user tries to provide as input. Commercial systems like Wii rely on a combination of sensors, besides the accelerometer, to decipher the gesture being performed. IR sensors inside the remote control, detect motion by tracking the relative movement of IR transmitters mounted on the display. Pressing hardware buttons may also signify the start of a movement; unfortunately, however, the development framework of the current test device prohibited access to this functionality. So before testing an "open-accelerometer" approach the authors resorted to a solution of "push-to-move" (similar to the analogy of "open-mike" and "push-to-talk").

In the push-to-move configuration, initiation of the gesture recognition was manually triggered by tapping anywhere on the tablet's screen (size of the screen: 7 in.). End of recording was done automatically after 2.5 s, which was selected empirically from previous studies. Figure 10(right) shows the average point that each user has chosen to tap in order to initiate the gesture recognition. From the one standard deviation of the points it can be suggested that users always tap on the same area. In a way this area represents a virtual button. Additionally, only participants who started with the gesture version (even-numbered id) picked a point outside the area of the previously located button bar and presented a more substantial deviation from the average point. Subjects marked with the odd-numbered id were probably biased by their first session with the button version.

In the second configuration the accelerometer was always on. In order to avoid the problems presented earlier a simple movement activity detector was implemented. The three-dimensional input signal (X, Y, Z) was merged into one acceleration magnitude. This was calculated by taking the Euclidean magnitude of the three individual values according to the formula: $\sqrt{x^2 + y^2 + z^2}$. The activity threshold was chosen empirically equal to $1.2\,\mathrm{M/s^2}$.

The reduced set of gestures under study provided high rates of correct gesture recognition. On average, a 94.5 % correct classification for push-to-move was obtained and 89.5 % for open-accelerometer ($t = 5.55$, $df = 7$, $p < 0.0001$). The box-plot of Fig. 11a shows the distribution among participants. Further analysis focused on alternative objective measurement around users' performance on the game per se. As user score the ratio between the correctly recognized sentences and the total number of sentences uttered is defined (in the present case 30). No significant differences in average scores was found between the three versions (89.26 % for button, 87.34 % for open-accel and 89.67 % for push-to-move), which is encouraging considering the challenges of using a new input modality for the first time (Fig. 11b). The similar score performance was also verified by the WER in the three versions. Using a 95 % confidence interval after a per-utterance bootstrap resampling [1] no significant difference was indicated in the three rates, specifically 92.3 % (C.I. 89.9–94.7 %) in the button version, 90.1 % (C.I. 88.2–93.6 %) in the open-accel and 91.2 % (C.I. 88.6–93.8 %) in the push-to-move. Concerning the average completion time of the experiments, the analysis reveals a difference of 3 min on average (button: 6.8 min and push-to-move: 9.77 min, $t = 6.61$, $df = 7$, $p < 0.0001$, Fig. 11c). At first glance this might seem quite high so further processing of the data was necessary in order to extract specific measurements that explain this difference.

As already mentioned, the experiment was organized around a specific pattern that users had to follow (next-help-speak). This pattern is considered as a turn in the experiment so that ideally participants had to perform 30 turns. First, the aim was to extract the average amount of

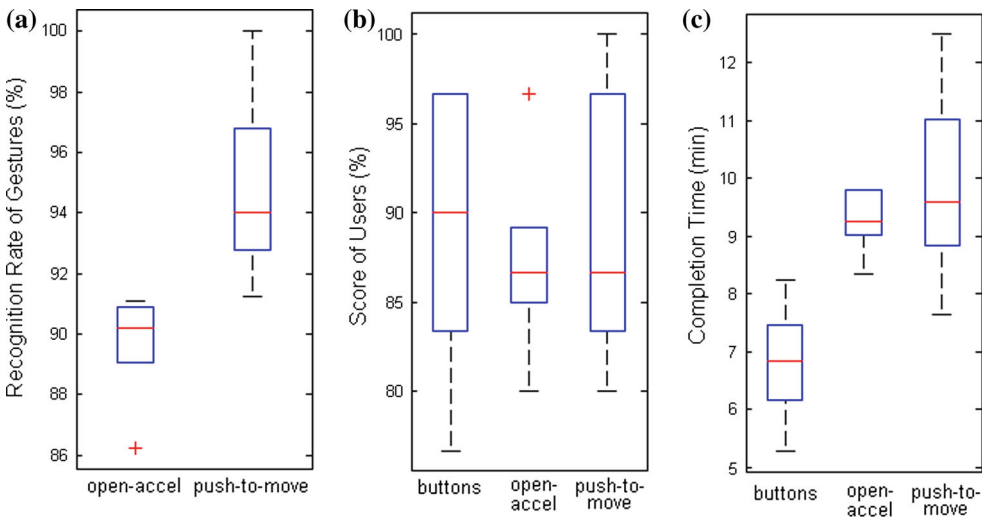

Fig. 11 **a** Box-plot of the gestures recognition rate, **b** Box-plots of the users score, **c** Box-plot of the completion time for each experiment. *Each box* is constructed to contain the 50 % of values closest to the mean, whereas the *horizontal line* represents the median value

time users spend on the turns in each of the three versions. The turn completion time is defined as the elapsed time between the dispatch of the next/previous message and the acquisition of the recognition result. The average time in the button version is 8.9, 12.6 s for open-accel and 13.5 s for the push-to-move ($F(3,127) = 102.83$, $p < 0.0001$). The specific difference (around 4 s) has an immediate explanation; the gesture processing step which takes roughly 1 s (1 s for the data acquisition and 7.6 ms for the recognition). In each turn duration this sums up to an accumulated overhead of 3 s. The corresponding probability density function is shown in Fig. 12a.

Continuing the analysis in respect to the difference in turn times, the time spent by users before interacting was examined. This quantity is named as "user time" defined as the elapsed time between the presentations of a prompt, a help example or a recognition result and user's interaction with the interface. During the "system time" the gesture is captured and recognized, the corresponding request is served and the result is presented. In Fig. 12b the user and system times in each turn are decomposed. The comparison of users' time between the two versions is an indication of how much more they had to think before interacting; in essence the additional mental load imposed on them. In Fig. 12c–f the plots that correspond to user time before the "next", "previous", "help" and "recognize" commands respectively are presented. As it can be observed, there are slight differences between the button and the gesture versions. A two-way ANOVA, identified significant main effects of interface type ($F(3,119) = 8.51$, $p < 0.001$) and gesture performed ($F(3,119) = 23.97$, $p < 0.001$) on the thinking time, showing that interacting with gestures does indeed impose a small mental overhead. A post-hoc Tukey's HSD ($p < 0.05$) pairwise comparison revealed the significant differences shown in Table 9.

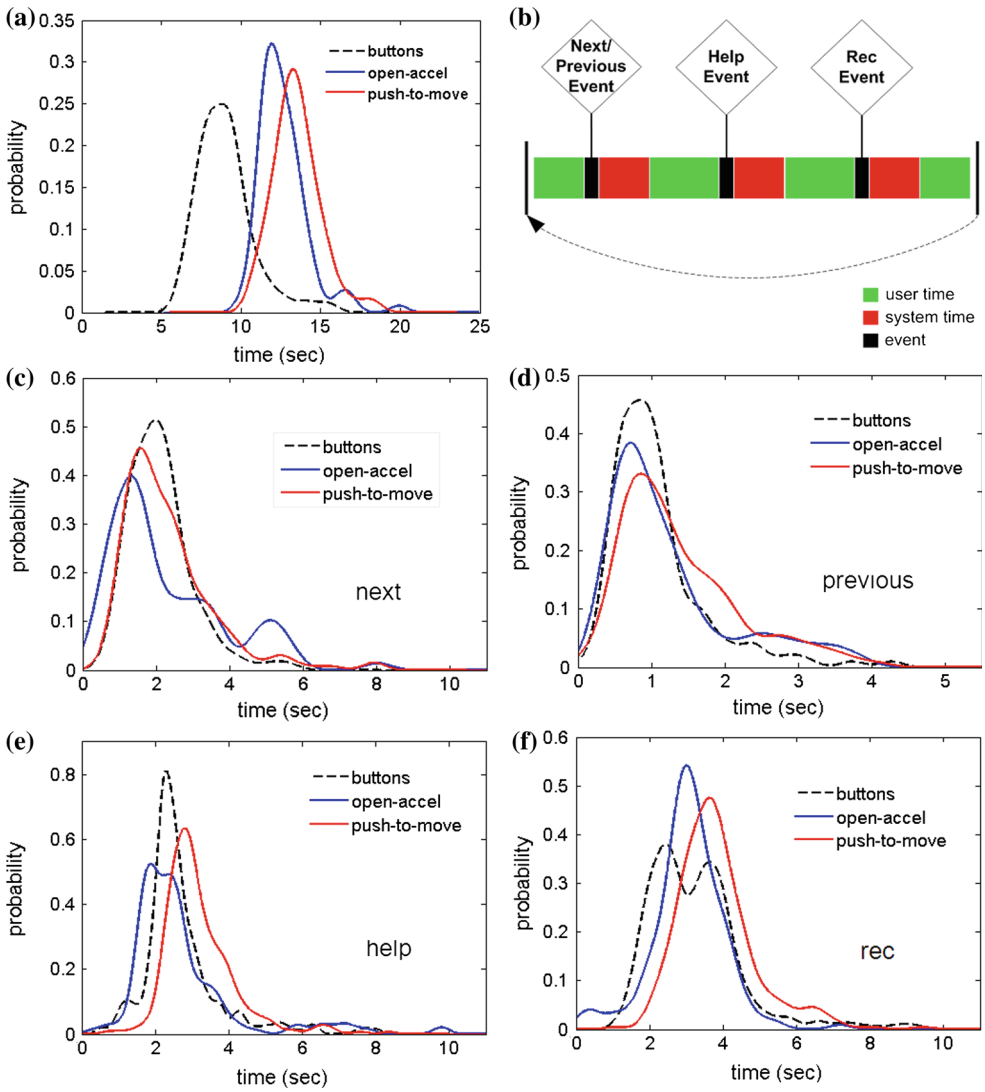

Fig. 12 **a** Probability density function of completion duration of each turn. **b** Decomposition of user and system times. **c** PDFs of thinking time before next, **d** previous, **e** help, **f** recognition gestures. Distributions approximated using kernel density functions

Table 9 Significance difference of interface versions and performed gestures in pairwise comparisons using Tukey's HSD test

	buttons	push-to-move	open-accel	
buttons		<0.001	0.29	
push-to-move			0.26	
open-accel				
	help	next	previous	recognize
help		<0.001	0.22	<0.001
next			0.43	<0.001
previous				<0.001
recognize				

According to these results, the mental effort is increased when the gesture is preceded by the extra action of tapping on the screen and not to the gesture per se. If different values for the button and the push-to-move are juxtaposed, the results are as follows: 2.1 versus 2.3 s (not statistically significant) when the input gesture is "next", 2.2 versus 2.8 s ($t = 3.9$, $df = 7$, $p < 0.01$) for "previous", 2.5 versus 3.2 s ($t = 2.8$, $df = 7$, $p < 0.01$) for "help" and 3.2 versus 3.8 s ($t = 2.09$, $df = 7$, $p < 0.01$) for initiating recognition.

The analysis concludes with the subjective evaluation of the interaction. In order to elicit the subjective opinion of participants a series of questions were asked in a paper-pencil questionnaire after the completion of each experiment. The answers were registered using a 1–10 Visual Analog Scale. Specifically, the aim was to assess issues like physical effort, concentration effort, performance of the system, user conformability and interaction preference. The average answers are presented in the radar plot of Fig. 13. As one can observe, participants report low levels or tiredness and medium levels of thinking effort. In accordance with the objective evaluation users corroborated the fact that the system worked well for both gesture and voice recognition. Concerning gesture recognition, users assigned a score of 8.4 for push-to-move and 6.6 for open-accel ($t = 4.26$, $df = 7$, $p < 0.0001$). Once again, the social acceptance of this type of interaction is verified with the low levels of users stating feeling uncomfortable while performing the gestures although it should be mentioned that the survey took place in an office environment with the presence of two observers at most. Users express a strong agreement that the gesture interface can help in certain situations and they have a very positive overall impression from the system. Finally, there is no evident consensus to which version users prefer most, although there is a tendency toward the button interface.

4.5 Accessibility for all

According to the World Report on Disability 2011[1], the number of disabled people in the world is presently estimated at around one billion, corresponding approximately to 15 % of the current world population. Similarly, the number of people older than 65 will reach 14 % of the world population in the next 30 years, rising to 1.4 billion by 2040 [41]. As stated in [38], disabled people prefer off-the-shelf devices over custom-made ones. Moreover, users with physical disabilities may prefer speech and hand gestures to keyboard or mouse to control computer systems [30]. The variety of accessibility techniques and the lack of interface consistency, however, force these users to learn new interactions models for every application they use. The authors strongly believe that the interaction paradigm provided in this work, where users can utilize a spoken dialogue application with their own gesture repertoire is a possible remedy for the aforementioned concerns.

In order to address possible issues related to different target groups, like users of lack fine motor control or vision-impaired users, three experiments were executed. Results will be presented, following an interview with a male subject aged 22 with mild cerebral palsy. Objectively, with no obvious communication disabilities, the person experiences kinetic problems that, besides others, prohibit efficient use of the keyboard. From the very first moment he was engaged in the conversation that lasted more than an hour. According to him, each person with cerebral palsy is a unique case, which makes the design of accessible interfaces a challenging task. He, as a regular user of dictation systems and other assistive technologies, had a very good idea of the hurdles posed in human computer interaction.

The first half of the interview concentrated on the introduction of the application and discussing common pitfalls encountered in other systems that should be avoided. Initially, the main concern posed by the participant was the poor results he experienced with other systems like eye blinking sensors. In this respect, the issue of the sensitivity in recognizing users' gestures was deemed of prime importance. The participant proposed to have a training phase before using the application, a feature that was already available in the system. Notwithstanding, the time and effort devoted for training should be the least possible given issues of physical and mental fatigue.

[1] http://www.who.int/.

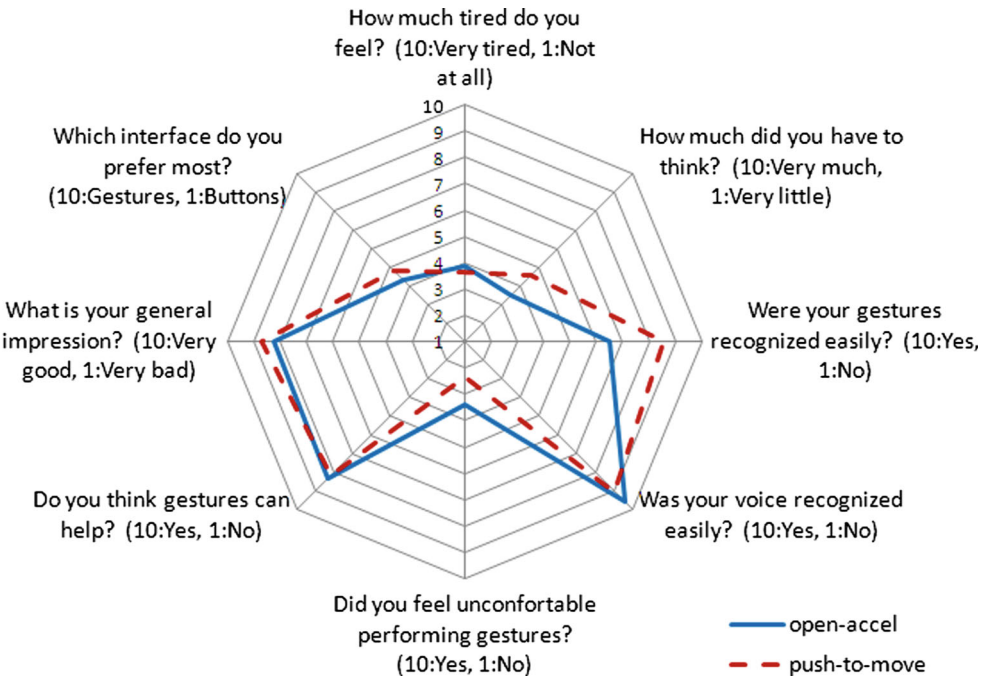

Fig. 13 Subjective evaluation results for the two gesture versions

The lack of many assistive systems to cover all the functionalities offered to regular users restricts their efficient usage and imposes the invention of alternative workarounds to perform them. Therefore, all these functionalities should be supported either by gestures or by other modalities (e.g. speech commands). Even before the proposed gestures for regular users were presented, issues related to social acceptability and discreteness of this type of interaction were discussed. The user stated that he would perform the gestures in front of everybody and in any place besides the pavement and the bar.

In the second phase of the interview he was asked to propose his own set of gestures and train the system. In order to facilitate the easy registration of the gestures, an interface that informed the user which one should perform each time was created. By utilizing a 3-s countdown counter the user was notified when to initiate the action. As explained to the participant, he could manipulate the device as he wished, in portrait or landscape orientation and by using one or both hands. He decided to hold the device with both hands in front of him (initial position) in portrait orientation and proposed the following gesture set:

1. **Next**. From the initial position move the device to the right.
2. **Previous**. From the initial position move the device to the left.
3. **Help**. From the initial position move the device upwards.
4. **Start speaking**. From the initial position move the device horizontally toward the torso.
5. **Stop speaking**. From the initial position move the device horizontally away from the torso.
6. **Abort**. From the initial position flip the device vertically parallel to the torso.

Each of the gestures had to be registered five times with the interface presented earlier. From the beginning of the registration process it was evident what the deficiencies of that approach were. The subject had difficulties coordinating his movements as dictated by the interface and considered the time allocated before the initiation of the action quite short (3 s). This miscoordination had a negative impact on the data provided, as sometimes the user executed the wrong gesture. More important, however, was the time he spent to execute a gesture that frequently surpassed the limit of 1 s in which the accelerometer was polled. The specific problems were reflected to the gesture recognition rate, as for the SVM case 74.29 % correct classification was obtained.

In order to quantify the energy of the acceleration signal, a non-disabled person was asked to execute the same gestures. Table 10 presents the RMS values in each of the axes and for each movement. The table provides an indication of the intensity of each gesture executed. The energy of the signal in the primary acceleration axis related to the gesture performed is depicted in bold. In combination with the standard deviation the user seems to have trouble executing the gestures intensively, something that was obvious during the experiment. Acceleration data were also acquired while the user was holding the device in front of him (initial position).

Table 10 Mean *X*- *Y*- *Z*- RMS and standard deviation value of each gesture signal

	next			previous			help		
	X	Y	Z	X	Y	Z	X	Y	Z
RMS	**.17**	.47	.96	**.2**	.43	.0.97	.12	.5	**.56**
sd	.07	.08	.13	.07	.08	.12	.04	.11	.19
RMS	**.31**	.31	1.04	**.31**	.25	1.04	.08	.56	**.88**
sd	.1	.008	.15	.01	.01	.01	.01	.01	.01
	start			stop			abort		
	X	Y	Z	X	Y	Z	X	Y	Z
RMS	.1	**.31**	1.02	.13	**.33**	1.01	.08	.82	**.73**
sd	.03	.08	.12	.04	.08	.16	.03	.22	.27
RMS	0.13	**.38**	1.04	.07	**.35**	1.04	.08	.66	**.96**
sd	.01	.01	0.1	.01	.01	.01	.01	.01	.01

The row with the gray background corresponds to the disabled person. RMS value for the principal acceleration axis is in bold

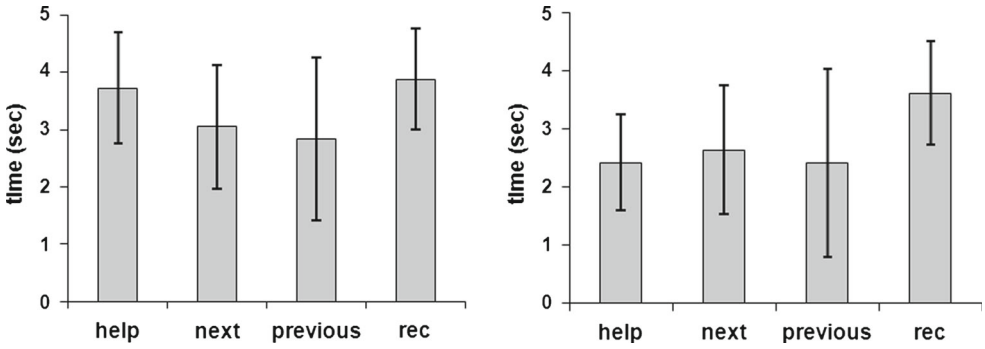

Fig. 14 Average thinking times for the user with cerebral palsy (*left*) and the blind subject (*right*). *Error bars* show one standard deviation

Spectral analysis did not show any indication of tremor that could influence the results.

The user proposed to combine voice commands and gestures, especially for picking list items. Another constructive remark was the lack of a "repeat" gesture that could facilitate the interaction. However, the deficiencies presented earlier prohibited efficient usage of the system and it was therefore decided to hold a second round of experiments after these issues had been resolved.

After introducing the movement activity detection component to the training interface the user was invited for a new experiment. This time another set of gestures was proposed, which was executed by holding the device in landscape orientation with the two hands (as a steering wheel). Gestures "next" and "previous" were performed by turning the wheel right and left respectively, and "help" by shaking the device right and left. To initiate recognition the tablet had to be brought close to the mouth and for stopping recognition the opposite; "abort" was signified by facing the screen display upwards.

In this case, the gesture recognition rate was 94 %, which shows that this set was well suited to the user's needs. The average turn time lasted 13.8 s (SD = 2.1 s) and the user achieved a score of 93 %. Figure 14(left) presents the average thinking times before each gesture. Similarly to non-disabled users the "recognize" movement imposes the higher mental effort as it must be combined with the speaking task. Finally, the accidental press of the hardware buttons of the tablet, located near the left palm of the user, caused temporal inconvenience.

The second experiment was conducted with a 25 year old blind female subject with 0.01 % vision capability. The interaction paradigm presented new challenges as the user had to be notified about the outcome of her gesture. For this reason a set of brief, distinctive sounds (earcons) was embedded to signify "next", "previous" and "recognize". After a "help" gesture the system started playing back the corresponding help prompt as before. Nonetheless, no feedback was provided about the recognition result (success or failure), a deficiency that should be addressed in a future experiment. During the training phase the registration of each movement started after a distinctive sound. However, the user was informed in advance which gesture to execute.

The gesture recognition rate was similar to the one for non-disabled users and equal to 89 %. The average turn time lasted 13.2 s (SD = 1.9 s) and the user achieved a score of 85 %. Figure 14(right) summarizes again the average thinking times before each gesture, which are comparable to the results of using the open-accel version presented earlier. With regard to the social acceptability of this type of interaction, the user did not state any concerns performing the gesture in front of different audiences neither in diverse environments. Some privacy issues were addressed as the user would prefer to get feedback with vibrations instead of earcons.

For the third experiment a female 65 year old subject, who had poor familiarity with technology and no previous exposure to similar systems was recruited. The participant was asked to use both the button and the gesture (open-

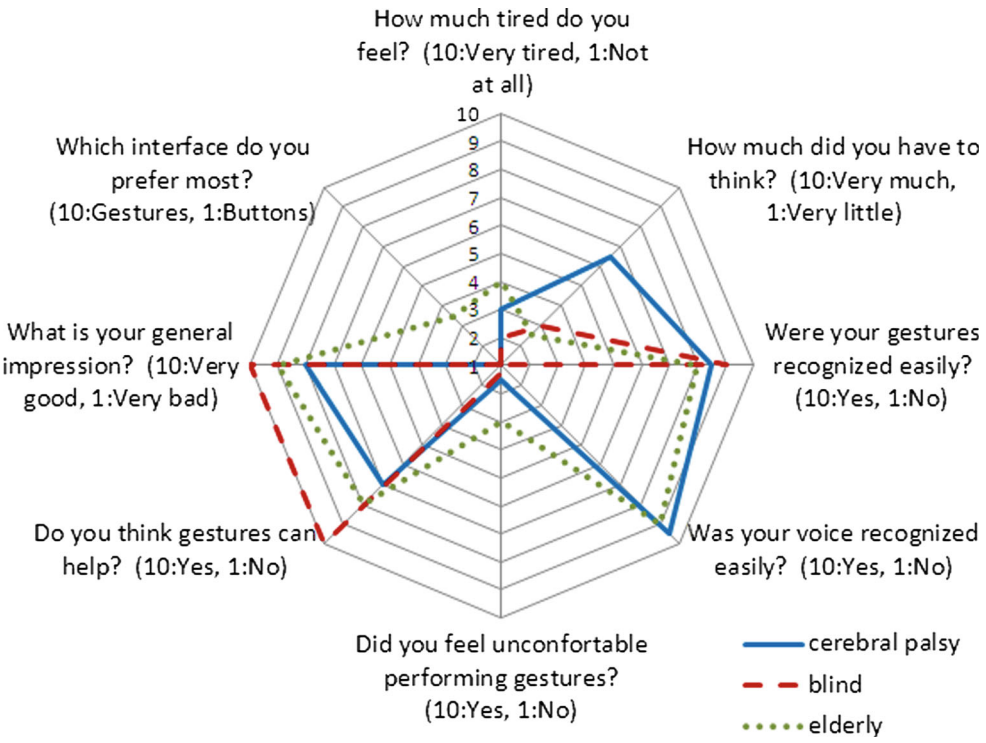

Fig. 15 Subjective evaluation results for the three subjects

accelerometer) interfaces. The aim of the study and the required tasks were explained as before and the subject started with the gesture version. Despite the fact that she did not express any concerns about the assigned task, the first reaction after holding the device was to replace her long distance glasses with the short distance ones. In general, the interaction was unhindered in both interfaces and real problems occurred only when the gesture recognition was unsuccessful. The user seemed to be preoccupied with performing each turn (next-help-speak) without really examining what was displayed on the screen. Even the different earcons associated with each gesture did not help a lot, as the user continued performing each step of the interaction pattern without checking the results of her actions. It was therefore needed to intervene when necessary, explain what the problem was and asked her to repeat the gesture. For this reason it was not possible to extract comparative results between the two interface versions. Finally, the correct gesture recognition rate was approximately 74 %, as the participant did not always perform them in a consistent way.

Figure 15 presents the subjective evaluation results. All participants were very positive about the already implemented system and its potential to help in certain situations. Neither of them expressed concerns or discomfort during its usage and all confirmed that it worked well. Moreover, the subject from the third experiment, having used two interfaces, seems to prefer the one containing buttons. Finally they all reported low levels of tiredness after 30 turns, although the user with cerebral palsy had to think more before performing a gesture.

5 Conclusions

This paper has described a prototype version of a gesture-driven spoken dialogue system hosted on a mobile tablet computer, and presented a series of evaluation tasks. Specifically, a concise and intuitively meaningful gesture set that can be used to trigger commands to any SDS has been introduced. A series of classification tests for this application task has also been performed. Guidelines for designing socially acceptable gesture interface were also provided. It has been illustrated that interacting with hand gestures imposes little physical and mental effort and results have been provided following interviews with a user with cerebral palsy, one blind user and an elderly person.

The proposed gesture set can be consider as a case study that may be interesting to designers that intend to embed motion sensing functionalities in their speech-enabled applications. Future extensions of this work include follow-up studies where subjects interact using their own set of gestures and also perform them in public settings. Investigation of more robust open-accelerometer techniques in combination with advanced gesture activity detection algorithms will exploit this idea to its full extent. More feedback from less studied target groups or from people with functional diversities would also be beneficial.

Finally, experimentation with other classification techniques or by combining different set of features could provide even more accurate results and more efficient usage of the device's resources.

Applications emanating from the game industry have made everyone aware of the potential of interfaces based on motion sensing; speech-enabled applications on mobile devices have only become common the last few years, and connections between the two technologies have not yet been widely discussed. It is surprising to see what rich synergies are available, which need to be explored further.

References

1. Bisani, M., Ney, H.: Bootstrap estimates for confidence intervals in ASR performance evaluation. In: Proceedings of the IEEE Conference on Acoustics, Speech, and Signal Processing (ICASSP), pp. 409–411. Montreal, Canada (2004)
2. Bolt, R.A.: Put-that-there: voice and gesture at the graphics interface. In: Proceedings of Computer Graphics and Interactive Techniques. Seattle, Washington, United States (1980)
3. Bouillon, P., Halimi, S., Rayner, M., Tsourakis, N.: Evaluating a web-based spoken translation game for learning domain language. In: Proceedings of the Fifth International Technology, Education and Development Conference. Valencia, Spain (2011)
4. Carbini, S., Delphin-Poulat, L., Perron, L., Viallet, J.E.: From a wizard of oz experiment to a real time speech and gesture multimodal interface. Signal Process. **86**(12), 3559–3577 (2006)
5. Cho, S.-J., Choi, E., Bang, W.-C., Yang, J., Sohn, J., Kim, D.Y., Lee, Y.-B., Kim, S.: Two-stage recognition of raw acceleration signals for 3D-gesture-understanding cell phones. In: 10th International Workshop on Frontiers in Handwriting Recognition (2006)
6. Choe, B., Min, J., Cho, S.: Online gesture recognition for user interface on accelerometer built-in mobile phones. In: Neural Information Processing, Models and Applications (2010)
7. Dong, L., Frank, E., Kramer, K.: Ensembles of balanced nested dichotomies for multi-class problems. In: PKDD, pp. 84–95 (2005)
8. Ferscha, A., Vogl, S., Emsenhuber, B., Wally, B.: Physical shortcuts for media remote controls. In: Proceedings of the 2nd International Conference on Intelligent Technologies For Interactive Entertainment (ICST). Brussels, Belgium (2007)
9. Fuchs, M., Tsourakis, N., Rayner, M.: A Lightweight scalable architecture for web deployment of multilingual spoken dialogue systems. In: Proceedings of the 8th International Conference on Language Resources and Evaluation (LREC) 2012. Istanbul, Turkey (2012)
10. Gama, J.: Functional trees. Mach. Learn. **55**, 219–250 (2004)
11. Hall, M, Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I.: The WEKA data mining software: an update. SIGKDD Explor. **11**(1), 10–18 (2009)
12. Hauptmann, A.G.: Speech and gestures for graphic image manipulation. ACM SIGCHI Bull. **20**, 241–245 (1989)
13. Joachims, T.: SVM-multiclass http://svmlight.joachims.org/svm_multiclass.html (2004)
14. Kane, S., Wobbrock, J.O., Ladner, R.: Usable gestures for blind people: understanding preference and performance. In: Proceedings of the Annual Conference on Human Factors in Computing Systems (2011)
15. Kauppila, M., Pirttikangas, S., Su, X., Riekki, J.: Accelerometer based gestural control of browser applications. In: International Workshop on Real Field Identification (RFId2007). In Conjunction with 4th International Symposium on Ubiquitous Computing Systems (UCS 2007), pp. 25–28 (2007)
16. Kela, J., Korpipää, P., Mantyjarvi, J., Kallio, S., Savino, G., Jozzo, L., Marca D.: Accelerometer-based gesture control for a design environment. In: Personal Ubiquitous Computing, July 2006, pp. 285–299 (2006)
17. Kristoffersen, S., Ljungberg, F.: Making place to make IT work: empirical explorations of HCI for mobile CSCW. In: Proceedings of the International ACM SIGGROUP Conference on Supporting Group Work, pp. 276-285. ACM, New York (1999)
18. Lee, K.-B., Kim J.-H., Hong K.-S.: An implementation of multimodal game interface based on pdas. In: Software Engineering Research, Management and Applications 2007. Proceedings of the 5th ACIS International Conference on, Busan, Korea, 2007, pp. 759–768 (2007)
19. Lee, Y., Kozar, K., Larsen, K.: The Technology Acceptance Model: Past, Present and Future. In: Communications of the ACM (Volume 12, Article 50), 2003, 752–780 (2003)
20. Lim, C.J, Pan, Y., Lee, J.: Human factors and design issues in multimodal (speech/gesture) interface. Int. J. Digit. Content Technol. Appl. **2**(1), 67–77 (2008)
21. Liu, J., Kavakli, M.: A survey of speech-hand gesture recognition for the development of multimodal interfaces in computer games. In: Proceedings of the IEEE International Conference on Multimedia and Expo (ICME), pp.1564–1569 (2010)
22. Liu, J., Wang, Z., Zhong, L., Wickramasuriya, J., Vasudevan, V.: uWave: Accelerometer-based personalized gesture recognition and its applications. In: Proceedings of IEEE International Conference Pervasive Computing and Communication (PerCom) (2009)
23. Martin, B.: Instance-Based Learning: Nearest Neighbor with Generalization. Master Thesis, University of Waikato, Hamilton, New Zealand (1995)
24. McGookin, D., Brewster, S., Jiang, W.: Investigating touchscreen accessibility for people with visual impairments. In: Proceedings of NordiCHI '08, pp. 298-307. ACM, New York (2008)
25. McNeill, D.: What Gestures Reveal About Thought. The University of Chicago Press, Chicago (1992)
26. Morganti, E., Angelini, L., Adami, A., Lalanne, D., Lorenzelli, L., Mugellini, E.: A smart watch with embedded sensors to recognize objects, grasps and forearm gestures. Procedia Eng **41**, 1169–1175 (2012)
27. Mustonen, T., Olkkonen, M., Hakkinen, J.: Examining mobile phone text legibility while walking. In: CHI'04 Extended Abstracts on Human Factors in Computing Systems (2004)
28. Neto, A., Duarte, C.: A study on the use of gestures for large displays. In: Proceedings of the 11th International Conference on Enterprise Information Systems (ICEIS) (2009)
29. Oviatt, S.: Multimodal interfaces. In: Jacko, J., Sears, A. (eds.) The Human-Computer Interaction Handbook, pp. 482–503. Lawrence Erlbaum Associates, Mahwah, New Jersey (2003)
30. Oviatt, S., Cohen, P., Wu, L., Duncan, L., Suhm, B., Bers, J., Holzman, T., Winograd, T., Landay, J., Larson, J.: Designing the user interface for multimodal speech and pen-based gesture applications: state-of-the-art systems and future research directions. Hum. Comput. Interact. **15**(4), 263–322 (2000)
31. Perakakis, M., Potamianos, A.: Multimodal system evaluation using modality efficiency and synergy metrics. In: Proceedings of the 10th International Conference on Multimodal Interfaces, Chania, Crete (2008)
32. Prasad, V.S.N., Kellokumpu, V., Davis, L.S.: Ballistic hand movements. In: Proceedings of AMDO'2006, pp.153–164 (2006)

33. Prekopcsak, Z.: Accelerometer based real-time gesture recognition. In: Proceedings of the 12th International Student Conference on Electrical Engineering, Prague, Czech Republic (2008)
34. Rico, J., Brewster, S.: Usable gestures for mobile interfaces: evaluating social acceptability. In: Proceedings of the 28th International Conference on Human Factors in Computing Systems (CHI '10), pp. 887–896. ACM, New York (2010)
35. Ronkainen, S., Haäkkila, J., Kaleva, S., Colley, A., Linjama, J.: Tap input as an embedded interaction method for mobile devices. In: Proceedings of TEI 2007, pp. 263–270. ACM Press, New York (2007)
36. Schlömer, T., Poppinga, B., Henze, N., Boll, S.: Gesture recognition with a Wii controller. In: Proceedings of TEI'08 Conference Tangible and Embedded Interaction, pp. 11–14 (2008)
37. Sears, A., Young, M.: Physical disabilities and computing technologies: an analysis of impairments. In: Jacko, J., Sears, A. (eds.) The Human-Computer Interaction Handbook, pp. 482–503. Lawrence Erlbaum Associates, Mahwah, New Jersey (2003)
38. Shinohara, K., Wobbrock, J.O.: In the shadow of misperception: assistive technology use and social interactions. In: Proceedings of CHI (2011)
39. Tanaka, K.: Next major application systems and key techniques in speech recognition technology. In: Proceedings of the IEEE Conference on Acoustics, Speech, and Signal Processing (ICASSP) (1998)
40. Turunen, M., Kallinen, A., Sanchez, I., Riekki, J., Hella, J., Olsson, T., Melto, A., Rajaniemi, J.-H., Hakulinen, J., Makinen, E., Valkama, P, Miettinen, T., Pyykkönen, M., Saloranta, T., Gilman, E., Raisamo, R.: Multimodal interaction with speech and physical touch interface in a media center application. In: Proceedings of the International Conference on Advances in Computer Entertainment Technology, New York, NY (2009)
41. U.S. Census Bureau: An Aging World, issued in June 2009
42. Wang, C., Seneff, S.: Automatic assessment of student translations for foreign language tutoring. In: Proceedings of NAACL/HLT 2007. Rochester, NY (2007)
43. Wobbrock, J.O., Wilson, A.D., Li, Y.: Gestures without libraries, toolkits or training: a $1 recognizer for user interface prototypes. In: Proceedings of the ACM Symposium User Interface Software and Technology (UIST) (2007)
44. 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)