A Corpus for a Gesture-Controlled Mobile Spoken Dialogue System

Nikos Tsourakis, Manny Rayner
ISSCO/TIM/FTI, University of Geneva, Switzerland
Boulevard du Pont-d'Arve, CH-1211 Genève 4, Switzerland
E-mail: {Nikolaos.Tsourakis, Emmanuel.Rayner}@unige.ch

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 through gestures. Situation-induced disabilities or real disabilities are determinant factors that motivate this type of interaction. In this paper we propose six concise and intuitively meaningful gestures that can be used to trigger the commands in any SDS. Using different machine learning techniques we achieve a classification error for the gesture patterns of less than 5%, and we also compare our own set of gestures to ones proposed by users. Finally, we examine the social acceptability of the specific interaction scheme and encounter high levels of acceptance for public use.

Keywords: Gestured-Controlled Mobile Applications, Gesture and Speech Interfaces, Gesture Classification

1. Introduction and motivation

According to (Hauptmann, 1989), 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 (Kristoffersen and Ljungberg, 1999) and as users, whilst mobile, perceive information differently (Mustonen et al., 2004). Although previous work provides some guidelines regarding gesture-based interfaces (Kane et al, 2011)(McGookin et al, 2008), little attention has been paid to the question of how to control spoken dialogue systems with gestures; most efforts have been directed towards seamlessly combining these two distinct input modalities in order to control multimodal interfaces (Liu and Kavakli, 2010), (Lim et al., 2008). A notable exception is the newly introduced feature of iPhone's Siri that permits the user to initiate speech recognition with a movement.

This paper describes an approach similar to that used by Siri but more elaborate, in which concise and intuitively meaningful gestures are used to trigger the core SDS commands. Specifically, we use a set of six gestures 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 we have incorporated these gestures in the mobile version of our CALL-SLT system (Bouillon et. al, 2011), 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.

Although our move in this direction was motivated by feedback from normally enabled 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. According to the World Report on Disability 2011 (http://www.who.int/), the number of disabled people in the world is presently estimated at around one billion, corresponding approximately to 15% of the current world population. The coordination required to use a normal button-controlled interface is experienced as challenging by many normally-enabled people, and would be beyond the reach of almost all users who experience problems with sight or fine motor skills.

On the other hand special kind 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" (Sears and Young, 2003) 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.

In contrast, we think it likely that the gesture-based interface like the one described here could be operated in many of these situations. If, for example, the device is strapped to the user's hand, 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. It would also be straightforward to add a "speech-only-output" mode which could be used even by completely blind people.

In this work, apart from introducing the gestures, we asked 8 users to perform and to evaluate them. Using machine learning techniques, we tried to quantify how well we can separate each gesture pattern and thus obtain a good estimate of what we can expect of a future deployed system. We also asked participants 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 our public appearance. Finally, we provide to the community a corpus of data gathered from users.

The rest of the paper is organized as follows. Section 2 describes the CALL-SLT gesture-based interface, and

Section 3 the data collection protocol. Section 4 presents a series of experiments designed to evaluate performance issues. The final section concludes.

2. A Gesture-Based Interface

CALL-SLT is a generic multilingual Open Source platform based on the "spoken translation game" idea of (Wang and Seneff, 2007). 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 (Bouillon et. al, 2011) and the top-level software architecture of the system in (Fuchs et al., 2012).

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. Figure 1 shows a screenshot of the GUI for the mobile version of the CALLSLT system.

Figure 1: CALL-SLT 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

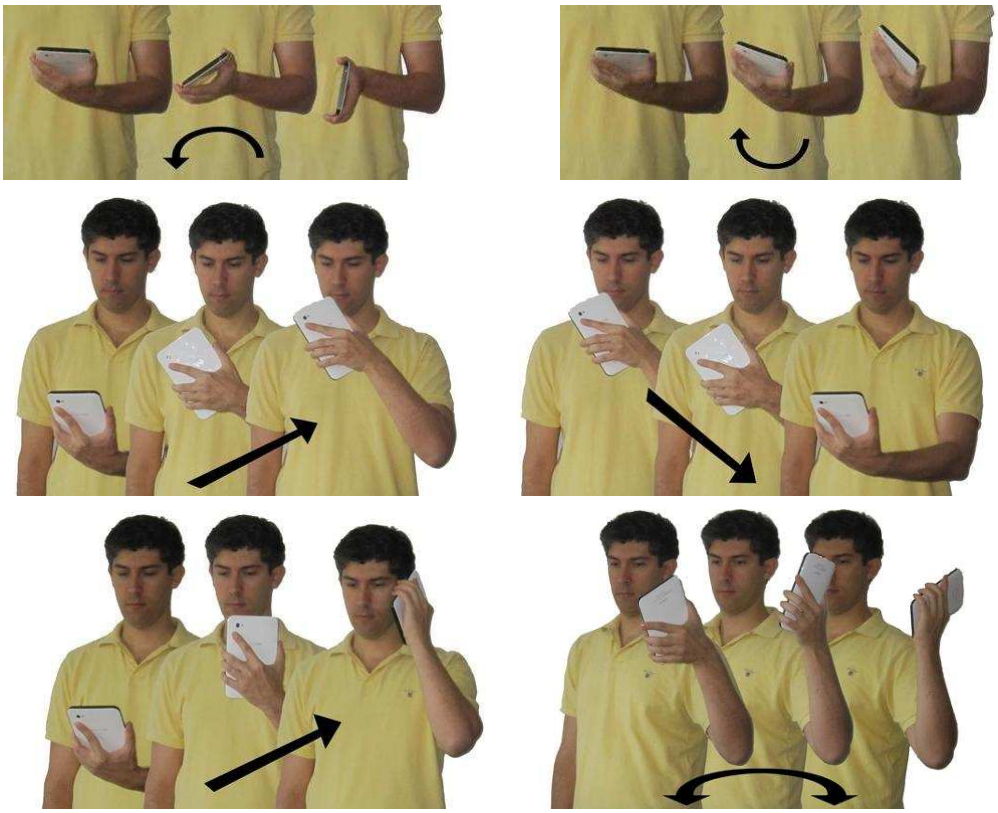

Figure 2: From left to right, bottom down next, previous, start recognize, stop recognize, help, abort

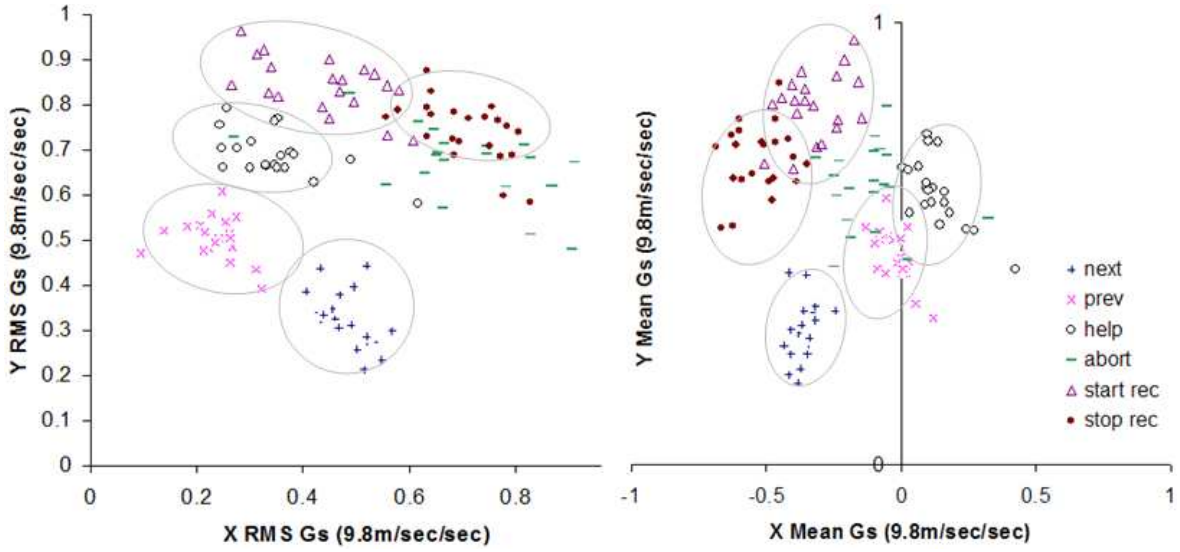

Figure 3: Separation of gestures in acceleration-space: RMS (left) and mean (right) values of the X and Y components of acceleration for Subject 1

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 one of the points 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, we have recently begun investigating the use of an interface which controls the key CALL-SLT functionalities using the intuitive gestures shown on Figure 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 degrees, 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.

3. Data collection

We used the Galaxy Tab's onboard accelerometer, 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. We collected 20 examples of each command from eight subjects, half male and half female, between 20 to 50 years old with higher academic education; half of them had no IT background. The eight right-handed subjects used the device as depicted in the diagram, holding it in their left hand while seated. The registration of each gesture was initiated after pressing a start button. This had the benefit that each interaction starts from the initial position.

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. We also collected similar data for eight common non-gesture conditions shown in Table 1.

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

Table 1: Non-gesture movements used in experiment

We extracted the mean and Root Mean Square (RMS) values for the X-, Y and Z-axis components, and used these six values as our features. The plots in Figure 3 show the data-points for the X-Y plane, tagged by gesture, for one of the subjects. Even with our very basic feature-space, Figure 3 suggests that the gestures should be easy to separate from each other.

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%

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

	Movements (gestures - nongestures)													
		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 3: Confusion matrix for the Support Vector Machine classifier

Classifier	Use the X, Y, Z acceleration frames (sampled every 50 msec for 1 sec)			
	Correctly Classified	Precision%	Recall	F-Measure
HMM	95.54%	96.36%	95.53%	95.34%

Table 4: Classification error (percentage) on gesture recognition using Hidden Markov Models

	Movements (gestures - nongestures)													
		a	b	C	d	e	F	g	H	i	J	K	l	m	N
a	Next	40	0	0	0	0	0	0	0	0	0	0	0	0	0
b	Previous	1	38	0	1	0	0	0	0	0	0	0	0	0	0
c	Help	0	1	37	1	1	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	1	39	0	0	0	0	0	0	0	0	0
f	Stop recognition	0	1	0	0	0	39	0	0	0	0	0	0	0	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	40	0	0
m	Descending	2	0	2	2	4	0	0	0	0	0	7	0	23	0
n	Walking	0	0	0	0	0	0	0	0	0	0	0	0	0	40

Table 5: Confusion matrix for the Hidden Markov Model classifier

4. Experiments

4.1 Gestures classification

In this subsection we present some results for gesture recognition. Different models have been proposed in the literature for this task, e.g. Dynamic Bayesian Networks (Cho et al., 2006), Support Vector Machines (Vitaladevuni et al., 2006) and Hidden Markov Models (Kauppila et al., 2007). Experimentation with some standard machine-learning algorithms confirmed this intuitive impression that the gestures could easily be separated from each other, and also showed that the gestures could be separated reasonably well from the non-gesture conditions. For each subject, we used 75% of the data (both gesture and nongesture) for training and 25% for testing. Classification was performed using Naive Bayes, Ensembles of Nested Dichotomies (Dong et al, 2005), 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 (Gama, 2004), Support Vector Machines (polynomial kernel and trade-off between training error and margin 5000) and Nearest-neighbor using non-nested generalized exemplars (Brent 1995).

The results of the different classification methods using the Weka Toolkit (Hall et al., 2009) are shown in Table 2, where we can see that most of the methods offer low error rates. Table 3 provides a better overview of the classification task for SVMs with the corresponding confusion matrix.

The methods presented earlier use features extracted from the sampled acceleration frames. 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 Hidden Markov Models 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 are a good candidate for gesture recognition.

The results shown in Table 4 were produced after training a left-to-right HMM with 6 states in the Weka Toolkit, for each gesture and user. Once again the confusion matrix in Table 5 shows of the responsible for most errors.

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 nongesture GUI and the presentation of a short video clip, they had to improvise gestures that would provide the same functionality. We tried to emphasize that the help is acoustic as well as visual and that one had to speak close to the microphone of the device. After the presentation of our own gesture repertoire, they 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 suggestion. The results of this survey are shown in Figure 4.

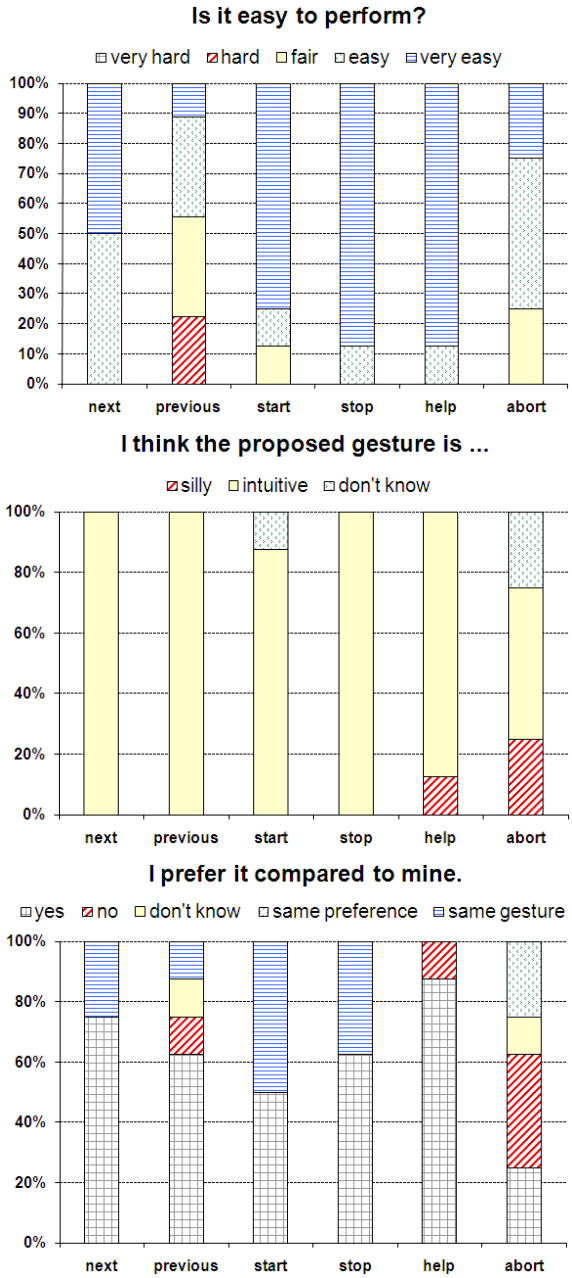

Figure 4: Charts of the easiness, impression and preference for each one of the proposed gestures

As we can observe, most of the subjects agree that the proposed gestures are easy to perform and are intuitive. They also prefer our set compared to theirs, with a small exception on the "abort" gesture. We believe that this has to do with the user's personal feelings concerning the specific movement. As matter of fact three of them had chosen the same gesture for "abort"; just flip the device, related to the metaphor of how to hang up a telephone set. According to another user this metaphor should also apply when you are using the system inside a car; you put the device down to signify "stop recognizing".

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

Table 6: Location and audience checklist

We have also encountered cultural differences as one subject proposed for "help" the hand gesture that signifies "question" for many Greeks (rotating the palm clockwise 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.

4.3 Social acceptability

As well as trying to determine how well gesture recognition works or if users prefer our set of gestures to theirs, another follow-up question was whether users would be willing to execute these gestures 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 with gestures. Notable exceptions are (Rico and Brewster, 2010) and (Ronkainen et al., 2007). Social factors have an influence on technology acceptance (Lee et al., 2003) so it is necessary to offer guidelines for the design and evaluation of socially acceptable gestures. We therefore continued our study by the 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 proposed gestures. The relevant checklist is shown in Table 6.

Their answers are summarized in Figure 5. As we can observe, our set of gestures receives a high level of acceptability even in public places. Pavements, public transportation and workplaces don't impose any usage limitations. On the other hand users seem reluctant to interact with gestures while driving; several of them made that this was for safety reasons. Concerning the audience of usage, there was universal positive agreement with a small exception of the "abort" gesture, which as we saw above was the most controversial one. Compared to the aforementioned studies the intuitiveness of our gestures for the specific application task has a beneficial impact on their social acceptability. During the design phase we tried to make them as simple as possible and also to exploit commonly acceptable interaction pattern. By putting the device close to the ear (help) or in front of the mouth (start recognition) we just reuse already accepted patterns. Likewise, the execution of "next" and "previous" commands resembles playing a mobile video game. Conversely, executing "abort" in public areas may attract undesired attention. In order to verify statistically the differences presented in Figure 5(down) we performed a significance test. The response variables of Table 6 can take only two possible outcomes (coded as 0 and 1) so we executed a Cochran's Q test. We found that there exist significant differences in gesture usage in diverse places ($X2(5) = 106.9$, $p < 0.001$). A pairwise comparison using continuity corrected McNemar's tests with Bonferroni correction revealed what the significant differences are, shown in Table 7.

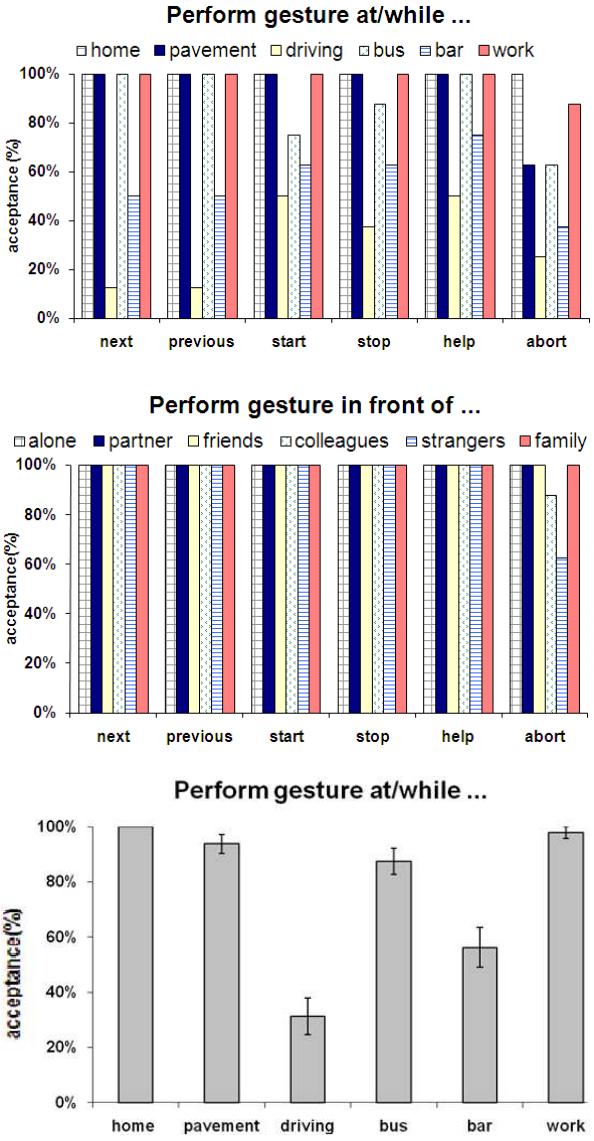

Figure 5: Average percentage of the gestures acceptability in different locations and in front of different people (error bars show one standard deviation)

	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						

Table 7: Significance difference of places in pairwise comparisons using continuity-corrected McNemar's tests with Bonferroni correction

5. Conclusions

We have described a prototype version of a speech-enabled conversation partner hosted on a mobile tablet computer, and presented a series of evaluation tasks. Specifically, we have introduced a concise and intuitively meaningful gesture set that can be used to trigger commands to any SDS. We also performed a series of classification tests for this application task and provided guidelines for designing socially acceptable gestures.

Possible future extensions of this work include follow-up studies where subjects interact using their own set of gestures and also perform testing in public settings. Feedback from less represented target groups (e.g. elderly people) would also be beneficial. Finally, experimentation with other classification techniques or by combining different set of features could provide 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; but speech-enabled applications on mobile devices have only become common within the last year or two, and connections between the two technologies have not yet been widely discussed. We are surprised to see what rich synergies are available, and plan to explore them further in the near future.

6. References

Brent, M. (1995). Instance-Based learning: Nearest Neighbor With Generalization. *Master Thesis*, University of Waikato, Hamilton, New Zealand.

Bouillon, P., Halimi, S., Rayner, M., Tsourakis, N. (2011). Evaluating a web-based spoken translation game for learning domain language. *Proceedings of the Fifth International Technology, Education and Development Conference*, Valencia, Spain.

Cho, S.-J., Choi, E., Bang, W.-C., Yang, J., Sohn, J., Kim, D.Y., Lee, Y.-B., Kim, S. (2006). Two-stage Recognition of Raw Acceleration Signals for 3D-Gesture-Understanding Cell Phones. In: *10th International Workshop on Frontiers in Handwriting Recognition*.

Dong, L., Frank, E., Kramer, S. (2005). Ensembles of Balanced Nested Dichotomies for Multi-class Problems. In: *PKDD*, 84-95.

Fuchs, M., Tsourakis, N., Rayner, M. (2012). A Lighweight Scalable Architecture For Web Deployment of Multilingual Spoken Dialogue Systems. *Proceedings of LREC 2012.*

Gama, J. (2004). *Functional Trees*. Machine Learning, vol. 55, pp. 219-250.

Hall, M, Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I. (2009). The WEKA Data Mining Software: An Update. *SIGKDD Explorations*, Volume 11, Issue 1.

Hauptmann, A.G. (1989). Speech and gestures for graphic image manipulation. *ACM SIGCHI Bulletin*, vol. 20, pp. 241– 245.

Kane, S., Wobbrock, J.O., Ladner, R. (2011). Usable gestures for blind people: understanding preference and performance. *Proceedings of the 2011 annual conference on Human factors in computing systems*.

Kauppila, M., Pirttikangas, S., Su, X., Riekki J. (2007). Accelerometer Based Gestural Control of Browser Applications. *In International Workshop on Real Field Identification (RFId2007). In conjunction* with Fourth *International Symposium on Ubiquitous Computing* Systems (UCS 2007), pp. 25-28.

Lee, Y., Kozar, K., Larsen, K. (2003). The Technology Acceptance Model: Past, Present and Future. *Communications of the ACM* (Volume 12, Article 50), 2003, 752-780.

Lim, C.J, Pan, Y., Lee, J. (2008). Human Factors and Design Issues in Multimodal (Speech/Gesture) Interface. *International Journal of Digital Content Technology and its Applications*, vol.2, no.1, pp. 67-77.

Liu, J., Kavakli, M. (2010). A survey of speech-hand gesture recognition for the development of multimodal interfaces in computer games, *2010 IEEE International Conference on Multimedia and Expo (ICME)*, pp.1564-1569.

McGookin, D., Brewster, S., Jiang, W. (2008). Investigating touchscreen accessibility for people with visual impairments. *Proc. NordiCHI '08, ACM (2008)*, 298 307.

Mustonen, T., Olkkonen, M., Hakkinen, J. (2004). Examining Mobile Phone Text Legibility while Walking. *CHI'04 extended abstracts on Human factors in computing systems*.

Rico, J., Brewster, S. (2010). Usable gestures for mobile interfaces: evaluating social acceptability. In *Proceedings of the 28th international conference on Human factors in computing systems (CHI '10)*. ACM, New York, NY, USA, 887-896.

Ronkainen, S., Hakkila, J., Kaleva, S., Colley, A., Linjama, J. (2007). Tap input as an embedded

interaction method for mobile devices. *In Proc. TEI 2007*, ACM Press, 263-270.

Sears, A., Young, M. (2003) Physical disabilities and computing technologies: An analysis of impairments. *In: The Human-Computer Interaction Handbook*, J. Jacko and A. Sears (eds). Mahwah, New Jersey: Lawrence Erlbaum Associates, pp. 482-503.

Vitaladevuni, S., Kellokumpu, V, Davis, L. (2006). Ballistic Hand Movements. In *Proceedings of AMDO 2006*. pp.153-164.