MASTER IN COMPUTER SCIENCE

JOG EASY

Multimodal Interfaces Project

2013

Haleh Chizari | Laura Daian | Kamakshi Sharma Jedele

Tutors:

Denis Lalanne | Elena Mugellini, | Jacques Bapst | Omar Abou Khaled

Contents

1. Introduction

For our project we developed an android application called JogEasy. The application has two uses two modalities – Gesture and Speech.

1.1 Requirement

The aim of the app was to enable touch free interaction for users. On discussing with frequent joggers we discovered a need gap in current jogging applications. Often when the user is jogging in winter, they wear gloves which make it difficult to use the app due to inability to effectively use the buttons on the app. Also, in summers when the user often gets sweaty while jogging he/she does not feel like touching the screen. In order to address these concerns we decided to develop an app with touch free, voice and gesture enabled interaction.

1.2 Description

The app itself has features such as:

- display of speed and altitude graphs
- track details with embedded Google maps
- storing track details in a database for later review
- time and distance

Through a combined use of voice and gesture recognition the recording of the track can be made to start and finish. The user should turn the phone to a 90 degrees angle such that the screen faces the mouth of the user and then say clearly ‘start’. At this point the app starts the GPS positioning and records the starting point on the map. Once the app has been started the user can put the device anywhere that is comfortable for him. The app records the track, speed, altitude and time taken for the jog. To finish the same gesture needs to performed and the user should say ‘stop’ clearly. Now the end of the track is noted on the map. The graphs as well as the track are displayed on the home screen.

Graphs and other details from the previous jogs can also be viewed on the app by clicking the icon on the app. The results are displayed in the form of date and time for each jogging session. A user can choose to rename a track, by long pressing and then selecting “Rename”. By short pressing on a session, graphs of speed and altitude, time, distance are displayed.

2. Resources

2.1 Eclipse

We have chosen Eclipse as the development environment. It provided us a good IDE, debugging tools and also easy accessibility to the project repository.

2.2 Android SDK

The Android Software Development Kit used in the project was version 4.2.2, and the Google API used for the map was version 17.

2.3 PocketSphinx SDK

For the voice recognition part we use PocketSphinx 0.8. We chose PocketSphinx because it is portable and easy to integrate into any application. It is light weighted, especially thought for mobile devices. It doesn't require Internet connection and you can use it anytime and anywhere.

We had a little bit of challenge at the beginning from the fact that it was a pretty new version and not yet very good documented for Windows users. At the moment of speaking a better documentation appeared under

https://sites.google.com/site/opiatefuchs/home/pocketsphinxandroiddemo

2.4 Google Maps

For displaying where the jogger has been running we used Google Maps. The version of the API is 17.

The track will color the map, according to the coordinates transmitted by the GPS every 2.5 seconds. The color used is blue. On the map we also have different kinds of pins. The colors that we used indicate where we have started (green), where we are now (red) and paused locations (yellow).

3. Modalities

We used in our application the following input modalities:

- Physical gesture(Touch and Movement)
- Speech

3.1 Voice

The application will take voice commands as input to start and stop the recording of a track. However, these commands must be used in combination of some gesture inputs. We used Pocket Sphinx for developing voice recognition. In the first prototype Google voice recognition was used. However, several problems were encountered.

- Need for constant internet to perform voice recognition
- Low adaptability to accents
- Slow recognition
- Many faulty recognition

As a result we decided to switch to PocketSphinx which overcame all these problems.

3.2 Gesture

We included the gesture modality for starting and stopping the recording. When the user moves his smartphone to a 90^{o} angles, and says “start”, then the application starts to record the tracking. For stopping the recording, the gesture is like the previous one and the voice command is “stop”.

We also added another feature: when the result screen is displayed (where the result of the jogging, speed time, graphs and the drawn track on the map, are shown) there are two possibilities to go back to the previous screen: we shake the smartphone or we press the back button.

As an alternative to noisy environments the user may use the gesture modality to start and stop the recording of a track. This is done by pressing the buttons present on the first screen of the app.

4. Care and Case

4.1 CARE

The CARE model focuses on the usability of a multimodal interaction. There are four ways to express the interaction: complementary, assignment, redundancy and equivalence.

In our application we use "***EQUIVALENCE***" modalities. The user can choose whether he/she wants to use the voice recognition or just the buttons. Also, when viewing the result of a track, the user can press the back button or can shake the phone. All these modalities are used exclusively.

When using the voice recognition there are several modalities taking into account, in one period of time. Gesture recognition is required when the user moves the phone vertical. Then the voice recognition is started and in the end another gesture recognition to stop the recording of the voice. All of these are used in a "***COMPLEMENTARY***" way, one after another in a certain period of time.

There are also commands "***ASSIGNMENT***". Like the ending of the app itself can be done only be pressing the back button, the view of the track list can only be open by pressing a button etc.

4.2 CASE

The CASE model indicates the use of different types of input in respect to the fusion modalities, as shown in Fig. 1. It is focused on the way the machine perceives the inputs.

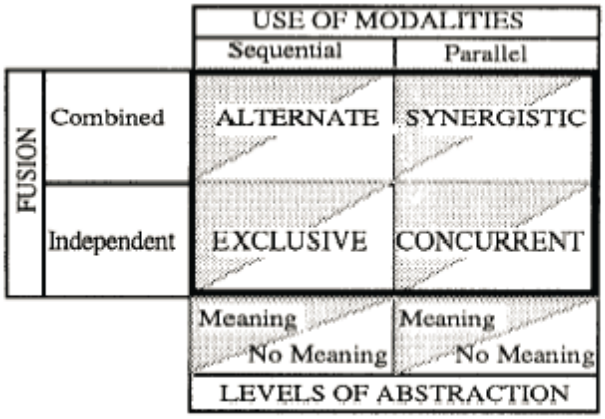

Figure 1: CASE Model

Our application uses the "***ALTERNATE***" method. The modalities used are sequential: gesture -> voice -> gesture. To use the voice recognition feature, the user needs to move the phone vertical to activate the recording of the voice. Then the commands like "Start" or "Stop" can be given, followed by the movement of the phone in some other position to stop recording.

5. User Evaluation

We performed a mix of qualitative and quantitative evaluations.

Target user

The table below gives a snapshot of users targeted for the evaluation:

Age	18-50
Gender	Male and Female
Type	Smartphone Users, known to record jogging
Nationality	Swiss

5.1 Quantitative Evaluation

While there are many parameters on the basis which the app can be evaluated, after some discussion it was decided to evaluate the app on the basis of the ability of a user to understand how to use the app by testing how long it took the user to perform simple tasks. We wanted to see if adding modalities to the app had resulted in too much complexity for use.

For our user evaluation we tested this hypothesis by comparing the time it took the user to perform the same task in our first prototype and our final design prototype.

To evaluate the app design we decided to test the app with some users in a controlled environment.

The user evaluation was performed with ten people. These people were divided into two groups and each group was unaware of the test activities being performed with the other group until after the test was over.

Test Group 1

To the five users in Test Group 1 design type 1 was presented for testing. Design type 1 was the basic app. The multimodal modalities of the app were not included and the basic interface was presented with only touch options.

Test Group 2

Design Type 2 was presented with all the modalities in function.

The procedure for the design evaluation is detailed below:

Step1. Introduce the App

All the basic functionality of the app was explained to the users through a small presentation and a walkthrough of the app.

Step2. Test Task

Here the users were given the real test task to see how proficient they are with the app. They were asked to use the app to record and view statistics with a 10 minute jog/walk. The time taken by them to start the app and to present the final statistics on the screen was recorded.

Step 3 T-test

An unpaired two tailed t- test was then performed on the timings to compare the findings. The timings were recorded in seconds. The data is unpaired since the two apps are different in terms of design and also the users are different. We decided to keep significance level as 0.5 to allow for a broader range of values.

Null Hypothesis: There is no significant difference between the times for understanding the two apps.

	TG1	TG2
1	34	37
2	21	32
3	37	31
4	32	38
5	40	39
mean	32.8	35.4
std dev	7.259477	3.646917

Using the Student two tailed t-test for independent samples with 95% significance:

$$t = \frac{\bar{X}_1 - \bar{X}_2}{S_{X_1 X_2} \cdot \sqrt{\frac{2}{n}}}$$

Sx1Sx2	5.744563
t	0.715626
dof	8
α	0.05
tcrit	2.306

Since the calculated t-value lies within the range of the critical value therefore the null hypothesis is not rejected and hence there is in fact no significant time difference between the user's performances between the two types.

After the t-test there was a brief discussion with the test users to gauge their preferences and prototype -2 emerged as a popular choice amongst users.

The question we are trying to answer is that if there is really any difference for the user between the two prototypes in terms of usage.

The results from the T-test were very close. They showed that the user did performed better with a simple app. It can be argued that since the test results were so close with time this difference will not matter. As the user gets to use the app this time difference will be overcome and the functionality will be the same. However, other factors such as final adoption and use, user preference, threat of competition etc. also need to be taken into account.

5.2 Qualitative Evaluation

After the test the other design of the app was revealed to the user and through a small interview we asked for feedback.

Upon a follow-up interview with users after both designs of the app were revealed, we received a mixed reaction.

Question 1

Would you use any of the apps for jogging if they are available for free?

90% agreed.

Question 2

Which design would you prefer to use?

- Some found the modalities exciting and would love to keep them, even if they use them regularly or not.
- Some found it a bit tiresome
- One user said that he would use the app if he would have to start it once before jogging and could interact with it after without having to touch it.

Question 3

What do you think about the idea of introducing modalities?

- Most liked the idea of being able to interact hands free with the app.
- One felt it was not so necessary and it worked just fine with touch.

6. Conclusion and Future Work

We were able to successfully develop an android application using two modalities – Gesture and Voice. The app recorded jogging tracks, along with speed, altitude graphs, time and distance. The track itself could be seen on Google maps. The modalities worked together to generate required results. The app received good feedback from the user on the whole. Test users seemed to be excited about using the app.

We used pocket sphinx for developing voice recognition rather than Google voice recognition as initial tests indicated pocket sphinx offered a much higher accuracy than Google. As the dictionary was created by us and had just 2 words such as – start and stop, no errors occurred during voice recognition. We tried testing the voice recognition with different accents and received a high success rate.

The gesture of turning the phone and then being able to talk was chosen over other gestures. This is so because while jogging the phone shakes quite a lot, it is possible that shakes may lead to starting of voice recognition, which would be counter-productive and lead to battery drainage and ineffective recording. The gesture of turning the phone to 90 degrees however, never happened while testing the app. Hence, we have also been able to achieve a high success rate with gesture recognition.

For future work, we would like to look at other possibilities of using a map, that might be more accurate as we discovered in our test runs that sometimes Google maps is unable to record small turns, which can lead to incorrect estimates. One feedback suggested that we keep the app

running and display statistic such as speed, time, distance on the screen, so that the user can check while jogging. We would also like to include this feature. We are also thinking of including another statistic a rough estimate of calories burned. Finally, we would like to make this app available on the play store for androids soon.

7. Bibliography

- "Multimodal Interfaces", Denis Lalanne, Elena Mugellini, Jacques Bapst, Omar Abou Khaled, Course Slides
- http://cmusphinx.sourceforge.net/2011/05/building-pocketsphinx-on-android/
- http://cmusphinx.sourceforge.net/wiki/tutoriallm # using_other_language_model_toolkits
- https://maps.google.com/