Augmented Reality Face Recognition For Mobile Devices

Gerry Hendrickx
Faculty of Engineering, Department of Computer Science
Katholieke Universiteit Leuven
Hendrickx.gerry@gmail.com

Abstract

This paper covers the current status and past work of the creation of an iPhone application. The application being created uses face recognition provided by Face.com [1] to recognize faces seen through the camera of the iPhone. Via augmented reality, the recognized faces will be named and enable the user to get information from different social networks. This paper will describe the research, the related work, the user interface prototyping and the current state of the implementation.

1. INTRODUCTION

Over the years, a lot of research has been done for augmented reality (AR). AR is a futuristic technology which augments a users view of the world. This can be done by using head mounted displays (HMDs) or smartphones. The technology can add extra information to the screen, based on geolocation or pattern detection. Geolocation apps have seen a big rise in the last years on the smartphones, with Layar as one of the leading apps [2]. Pattern detection apps are not yet as common, so they still offer a lot of unexplored possibilities. This paper describes the process of creating a pattern detection app for the iPhone. It will use face recognition to offer the user extra information about the persons seen through the camera. This extra information will come from various social networks, the most important being Facebook, Twitter and Google+. Its goal is to offer users fast access to online data and information that is publicly available on the internet. This can be used in a private context, to enhance conversations and find common ground with your discussion partner, or in an academic context, to be able to easily find information, like slides or publications, of the speaker at an event you’re attending. The app will be created for iOS because the SDK offers a built in face detection mechanism since iOS5 [3]. This feature will drastically simplify the tracking of the faces. Android, the other option, does not have this built in feature, so the choise wasn’t hard to make.

2. OBJECTIVE

A brainstorm and a poll on Facebook resulted in a list of the wanted requirements for the face recognition app. First of all the application should work fast. Holding your smartphone up and pointed to another person is quite cumbersome, so the faster the recognition works, the smaller this problem becomes. Another requirement is privacy. A second Facebook poll revealed that out of 34 voters, 14 did not like the idea of them being recognized on the streets. There was a strong need for privacy, so looking at the Face.com API, the following policy was decided upon: Face.com allows a user access to all of his Facebook friends. The app could be limited to this, but the need to recognize your Facebook friends is lower than to recognize strangers. Face.com also allows private namespaces per app, which means an app can create its own domain in which all the users of the app are stored. If a namespace is used, users of the app would be able to recognize other users. The general policy will be: if you use the app, you can be recognized. An eye for an eye. The functionality requirements from the poll and brainstorm and thus the goals to achieve are the following:

- Detection and recognition of all the faces in the field of view.
- Once recognized, the name and other options (like available social networks) will appear on screen with the face.
- Contact info will be fetched from the networks and can be saved in the contacts app.
- Quick access to all social networks will be available, along with some basic information such as the last status update/tweet.
Information available will differ from person to person. To add an extra layer of privacy settings, a user will be able to link his networks and choose to enable or disable them. When the user gets recognized, only his enabled networks will show up.

3. RELATED WORK AND RESEARCH

3.1. Related Work

For this master thesis we went to search for related mobile applications. No application was found that does exactly the same as in this project. However, some similar were found.

- **Viewdle**: Viewdle is a company focusing on face recognition. They have several projects ongoing, and have already created an application for Android, called Viewdle Social Camera. It recognizes faces in images based on Facebook. Viewdle has a face recognition iOS SDK available, but after contacting the company multiple times, no reply was received and we moved on to Face.com.

- **Animetrics**: Animetrics created apps to help government and law enforcing agencies. It has multiple products like FaceR MobileID, which can be used to get the names and percentage of the match of any random person. FaceR CredentialME can be used for authentication on your own smartphone. It recognizes your face and if it matches, it unlocks your data. Animetrics also focuses on home security face recognition. However they do not seem to have an API, which is a pity because their technology seems promising.

- **TAT Augmented ID**: TAT Augmented ID is a concept app. Basically it’s exactly the idea from this article. It recognizes faces in real time and uses augmented reality to display icons around the face. The concept is the same, but the resulting user interface is different.

Another non-commercial related work is a masterthesis of 2010 at the Katholieke Universiteit Leuven [10]. The author created a HMD-based application to recognize faces and get information. From his work we learned that HMD’s are not the ideal practical setup (his app required a backpack with a laptop and heavy headgear), and that the technology used (OpenGL) is a cumbersome way to develop. Using iOS simplifies these aspects. The author used Face.com as face recognition API and was very satisfied with it.

3.2. Face Recognition API’s

A couple of promising face recognition API’s were found and compared in order to find the one most suited to our goals. A quick summary of the positive and negative points:

- **Viewdle**: As said above, we’ve tried to contact Viewdle to get more information about the API. Sadly, they did not respond, therefore Viewdle is no option.

- **Face.com**: Face.com offers a REST API and is well documented. It offers Facebook and Twitter integration and a private namespace that will help with the privacy concerns. There is a rate limit on the free version of Face.com.

- **Betaface**: The only API to work with images and video. However, it is Windows only, hasn’t been used with iOS yet and is not free.

- **PittPatt**: PittPatt was a very promising service, but sadly it got acquired by Google. The service cannot be used at this time.

It seemed that Face.com is the only option and luckily the best one found. It has an iOS SDK and social media integration, which are both very useful in the context of this application.

4. PAPER PROTOTYPING

Paper prototyping is the process of designing the user interface based on quick drawings of all the different parts of the UI. By doing this with paper parts, it becomes easy to quickly evaluate and adapt the interface. The prototyping phase consisted of 3 iterations: the interface decision, the interface evaluation and the expert opinion.

4.1. Phase one: interface deciding

The first phase of the paper prototyping was to decide which of the 3 possible interfaces would be used. The interfaces were:

- **Interface 1**: A box of information attached to the head of the recognized person. This is the best use of augmented reality, but you have to keep your phone pointed to the person in order to be able to read his information.

- **Interface 2**: A sidebar with information, which takes about 1/4th of the screen. This way, users
4.2.1. Usefulness.

The overall results were good. People seem to understand why the app is useful and it does what they would expect. The scores on whether it meets the need of the users were divided, because some users (especially the ones with some privacy concerns or without a smartphone) did not see the usefulness of the application. However, the target audience (the smartphone users) did see its usefulness, resulting in higher scores in this section.

4.2.2. Ease of use. From the ease of use questions the need to add or rearrange buttons became clear. Users complained about the number of screen transitions it took to get from one screen to somewhere else in the application. This calls for more buttons, to ease the navigation. For instance, home button to go to the main screen will be added on several other screens, instead of having to navigate back through all previous screens. Using the application was effortless for all iPhone users, because the user interface was built using the standard iOS interface parts.

4.2.3. Ease of learning. None of the ease of learning questions scored below 5 on the 7-point Likert rating scale. This is also due to the standard iOS interface, which is developed by Apple to be easy to work with and easy to learn.

4.2.4. Satisfaction. Most people were satisfied with the functionality offered by the application and how it was presented in the user interface. Especially the iPhone users were enthusiastic, calling it a innovative, futuristic application. Non-smartphone users were more skeptical and did not see the need for such an application. Aside from this, the application was fun to use and the target audience was satisfied.

4.2.5. Positive and negative aspects. The users were asked to give the positive and negative aspects of the application. The positive aspects were the iOS-style of working and the functionality and concept of the application. The negative aspects were more user interface related, such as not enough home buttons, and the cumbersome way to indicate a false positive. This button was placed on the full screen information window of a person. Everybody agreed that this was too late, because all the data of the wrongfully tagged person would then be displayed. So the incorrect tag-button should be placed on the camera view. Some useful ideas like enabling the user to follow a recognized person on Twitter were suggested.

4.3. Phase three: expert opinions

For this phase, the promoter and tutors of the thesis were used in the paper prototype test. They all have extensive experience in the field of human-computer in-
teraction and can thus be seen as experts. They took the tests and gave their opinion on several aspects of the program. A small summary:

- There were concerns about the image quality of the different iPhones. Tests should be done to test from what distance a person can be recognized.

- The app should be modular enough. In a rapidly evolving 2.0 world, social network may need to be added or deleted from the application. If all the networks are implemented as modules, this will be a simpler task.

- The incorrect tag-button could be implemented in the same way as the iPhoto app asks the user if an image is tagged correctly.

- The social network information should not just be static info. The user should be able to interact directly from the application. If this is not possible, it would be better to refer the user directly to the Facebook or Twitter app.

- More info could be displayed in the full screen information window. Instead of showing links to all networks, the general information about the person could already be displayed there.

When asked which social networks they would like to see in the application, nearly everybody said Facebook, Twitter and Google+. In an academic context, they would like to see Mendeley and Slideshare.

5. IMPLEMENTATION

With the results of all paper prototyping test being positive, the next step is the implementation. The app is currently in development, and a small base is working. The main focus so far is on the crucial functionality, the face recognition. It is important to get this part up and running as fast as possible, because the entire application depends on it. So far the application is able to track faces using the iOS5 face detection. A temporary box frames the faces and follows them as the camera or person moves. This functionality could be used to test the quality of the face detection API. As you can see in figure 1, the face detection algorithm of iOS5 can detect multiple faces at ones, and in such depth that the smallest recognized face is barely bigger than a button. These are great results, because the algorithm appeared to be fast and reliable and detailed enough for our purpose. Detection of even smaller faces is not necessary, because the boxes will become harder to click if they are smaller than a button.

The next step was the face recognition. This is handled by Face.com. An iOS SDK is available on the website [11]. This SDK contains the functionality to send images to the Face.com servers, and receive a JSON response with the recognized faces. It also covers the necessary Facebook login, as Face.com requires the user to log in using his Facebook account. This login is only needed one time. One problem was that Face.com only accepts images, not video. To be able to get the face recognition working as fast as possible, a recognize-button was added to the camera view. Once clicked, a snapshot is taken with the camera. This snapshot is send to the servers of Face.com and analyzed for different faces. The JSON response gets parsed and the percentage of the match and the list of best matches can be fetched from it. At the moment, only one face can be recognized at the same time, because there is no way to check which part of the response should be matched to which face on the camera. This is thus temporarily solved by limiting the program to one face at a time. Figure 2 shows the current status of the application. A face is recognized and its Facebook ID is printed above.
6. NEXT STEPS AND FUTURE PROBLEMS

The next step in development is the further creation of the user interface. Now that we have a basic implementation of the main functionality, it is important to finish the outlines of the application. This way, a dummy implementation of several screens can be used to test the interface in several iterations using the digital prototype. When these tests happen, the underlying functionality can be extended and implemented in the background. This way, we can combine the two tasks. If the full underlying functionality would be implemented first, the user tests would start too late. In case big UI problems arise, it would be better to catch them early in the process.

Several big problems need to be solved. The biggest being the matching of faces detected by iOS5 face detection and the faces recognized by Face.com. Because Face.com recognizes faces by using images, a way needs to be found to match these results to the faces on screen. If the user moves the camera to other people after pressing the recognize-button, the results from Face.com will not match the faces on screen. The solution in mind is to use an algorithm to match the Face.com results with the face detection based on proportions. If we succeed in finding a correlation between for instance the eyes and the nose of a person in both services, it should be possible to find which detected face matches which Face.com result.

Another smaller problem is the use of the Face.com SDK. It has a limited Facebook graph API built into it. However, this API can not be used to fetch the name of an ID or to get status updates. Therefore the real Facebook iOS SDK should be used. To prevent the app from working with two separate API’s, the Face.com API needs to be adapted so that it uses the real Facebook iOS SDK instead of the limited graph API.

7. CONCLUSION

This masterthesis is still a work in progress. We already have good results from paper prototyping, and the core of the application has already been implemented. In the following months, some problems will have to be solved and a lot of user testing is still required to make the application match its goal: a fast, new way to discover people using the newest technologies and networks.

References

[7] http://www.youtube.com/watch?v=ib0pMeg1UN0