FaceSpeaker
Wearable Face Recognition Device for the blind
Information Science BSC thesis Project Report

T.A. in 't Veld
Information science student at Utrecht university
student number *3139549*
Thesis Supervisor: dr. ir. R.J. Beun

January 27, 2014

Abstract

The FaceSpeaker project developed a prototype wearable face recognition device which supports visually impaired users during social interactions by automatically identifying their acquaintances. The prototype is based on a laptop worn in a backpack, running the open source FaceSpeaker software. The user controls the device using a small Bluetooth keyboard. The user wears camera glasses containing a tiny unobtrusive camera. The FaceSpeaker software speaks an enrolled person's name when his face comes in view of the camera.

This paper provides an overview of comparable research efforts and describes the FaceSpeaker prototype. It analyzes the context in which FaceSpeaker will be used, and demonstrates how the device would benefit a user in the context of a small academic conference. Various requirements and design recommendations for the device are inferred.

Contents

1 Introduction

Suppose you regularly visit your favorite bar with a large group of friends. Now try to imagine heading to the bar blindfolded next time. Hopefully one of your friends is aware of your blindfolding experiment and comes to collect you at the door, because you'd have little hope of finding the group in the crowd otherwise. So he says "hi". Who is he? Hopefully you have a good memory for voices, or you might just follow a complete stranger. But let's suppose you ascertain his identity and he takes you to the table where your group is sitting. You take a chair. What next? Who is present tonight anyway? You just say "hi" to the friend sitting next to you - let's again assume you hear his voice and find out he is Bob. You have heard an ugly roomer about Alice, and since you don't expect her to be present you tell it to Bob. But then you hear a very angry voice - Alice was sitting quietly at another table several meters from you! Later when your group mixes in with the rest of the crowd, you get into a conversation with a rather nice member of your opposite gender. At some point (s)he heads off to the bathroom. (S)he likes you but is a bit shy and expects you to resume the conversation. But you didn't see him/ her returning, so you disappoint each other and are both out of luck!

The above thought experiment illustrates the disadvantages visually impaired people face during social interactions due to their inability to identify people around them. Part of the solution to help alleviate such issues might be an assistive device which could identify people around the user. The goal of the FaceSpeaker project is to design a wearable face recognition device which can be trained to recognize the user's acquaintances and convey an acquaintance's identity to the user when the acquaintance comes in view of the camera.

A working prototype was developed. The user wears camera glasses which are connected to a laptop computer carried in a backpack. When the FaceSpeaker software detects a person in view of the camera, it speaks that person's name (if it was previously trained to recognize that person) or reports it has detected an unknown face. The FaceSpeaker software powering this device is free, open source software. This means users can build their own face recognition device and other (research) projects can take advantage of it.

At first glance, the FaceSpeaker concept is very easy to understand. It is just a device which can be trained to unobtrusively identify the user's acquaintances. What makes the design complicated is not the technology. The technology is cutting edge to be sure, but as demonstrated by the FaceSpeaker prototype and various related projects all the necessary technology is available as of 2013.

What makes the design challenging is the fact the device will be used in contexts involving social interactions. Such interactions are governed by a set of rather subjective codes, norms and values which are applied almost subconsciously. The main goal of this paper is to propose a design which will be functional and socially acceptable in such contexts.

1.1 Paper Overview

This paper is roughly divided into 3 parts. In the first part, the context surrounding the FaceSpeaker concept is described. Chapter 2 provides a comprehensive overview of research efforts related to the FaceSpeaker project. In chapter 3 a PACT (people, activities, contexts, technologies) analysis is carried

out. This analysis serves to highlight factors related to each of the four PACT elements which will be taken into account when designing FaceSpeaker, and explores how those factors will vary across different settings in which FaceSpeaker will be used. The first part is concluded by chapter 4, which infers the main requirements for the FaceSpeaker design (unobtrusiveness and economizing on user attention) from the PACT analysis and the literature study.

The second part describes the prototype. Developing the prototype served to gain an understanding of the technical issues involved in designing the FaceSpeaker device. The prototype provides a useful "proof of concept" and is an effective tool to demonstrate the FaceSpeaker concept. Simple field trials were performed using the prototype, which served to elicit user feedback and highlight important issues relevant to the FaceSpeaker design process. Chapter 5 provides an overview of the prototype. It describes the hardware used and provides a high level overview of the technical details surrounding FaceSpeaker's development process. Finally the face recognition algorithms used by FaceSpeaker are discussed. Because technical issues surrounding software development and face recognition algorithms are not the focus of this study, the technical sections describe the issues involved on an extremely high level and leave out most technical details. Readers interested in technical details should consult the cited references, and programmers interested in FaceSpeaker's inner workings can obtain the full source code from the facespeaker.org website. For interested readers, appendix A provides a detailed description of the research involved in finding a camera for the FaceSpeaker prototype and appendix C contains the full user manual for the FaceSpeaker software (which may be downloaded from the facespeaker.org website). Another useful way to better understand FaceSpeaker's design is to watch the demonstration video on the facespeaker.org website.

The third part of this paper elaborates on the details of the FaceSpeaker design. Chapter 6 introduces the small conference scenario, which places the FaceSpeaker device in a concrete context by illustrating how the device supports a user in the context of a small academic conference. Chapter 7 discusses the procedure for enrolling people into FaceSpeaker's database. Chapter 8 discusses how FaceSpeaker should be triggered to identify people and how an identified person's identity should be conveyed to the user. The chapters provide various requirements and design recommendations, motivated by the context analysis as well as experiences gathered while developing and field testing the prototype. Finally, chapter 9 provides a quick summary of the conclusions drawn from the FaceSpeaker study.

1.2 facespeaker.org Project Website

A project website for the FaceSpeaker project is available at `http://www.facespeaker.org`. On this website, a demonstration video of the FaceSpeaker software is available and the software (including source code) may be downloaded. The website contains various other complementary project materials, and updates about (a follow-up to) the FaceSpeaker project will be posted to the homepage.

1.3 Problem Definition

This section provides an elaborate description of the problem which the FaceSpeaker project aims to help address. Face recognition is the primary method used by humans to identify each other. Visually impaired individuals have limited or no ability to recognize faces. There are alternative means to identify people such as voice recognition or recognizing physical characteristics like hair color or stature. But no combination of such alternative means can approach the speed and accuracy of normal face recognition ability.

Therefore, people who cannot recognize faces have a limited ability to rapidly identify other persons. An inability to rapidly identify other persons can cause many social problems including misunderstandings, social exclusion, limited self-confidence or safety hazards. For example, blind individuals don't spot acquaintances in a group or on the street. Therefore they may not greet acquaintances or initiate conversation. This can cause them to miss social opportunities, and it can be incorrectly perceived as lack of interest or rudeness inhibiting their social integration even further.

In summary, persons who cannot recognize faces face a major disadvantage in social interactions. The FaceSpeaker project aims to design a prototype face recognition device which will enable users to rapidly identify acquaintances, thereby helping users to alleviate the problems associated with their inability to rapidly identify other persons.

2 Related Projects

This chapter explores projects related to FaceSpeaker. There is an enormous body of research in the face recognition field, and a device to help users covertly recognize people around them has been widely proposed. Only a handful of projects have actually attempted to build such a system, and as of the start of the FaceSpeaker project in early 2013 none of those systems had yielded published software let alone marketable products. However, the number of projects leveraging computer vision technology to support the visually impaired community is currently growing rapidly as the necessary technology is maturing. Indeed, during the FaceSpeaker project the Israeli company Orcam [27] released a product leveraging computer vision to support partially sighted users. Functions of this device will include text, object and face recognition.

2.1 Similar Devices

A couple of research teams and companies have developed or proposed devices similar to FaceSpeaker. A student team at the University of Maryland has designed and developed a prototype face and expression recognition device for visually impaired users [3]. The prototype was similar to the FaceSpeaker device, utilizing a camera mounted atop a long cane connected to a laptop worn in a backpack. Besides face recognition their prototype performed facial expression recognition (e.g. telling the user if a person was smiling or looking sad). Their face identity recognition worked quite reliably and yielded positive feedback from testers. Facial expression recognition still poses major technical challenges. This project provided a lot of inspiration and material to the FaceSpeaker project.

Another research team at Arizona State University developed the iCare social interaction assistant [17]. In 2005, their starting point was a face recognition device [16]. Like the FaceSpeaker prototype, their system was based on camera glasses attached to a laptop worn in a backpack. Their system worked fairly well, at least under laboratory conditions. They proceeded to add other functionality to the system. In 2008 they performed a systematic requirements analysis for this device [15], mainly to find out which functionality beyond face recognition is most valued by potential users and to find some high level requirements for the proposed device. They proceeded to publish several articles around this concept. They addressed the issue of people localization [10], and conveyed interpersonal distance by using tactile rhythm [21]. Finally they worked on expanding the prototype to help blind users avoid socially disruptive stereotypic body mannerisms and attempted to have the prototype convey facial expressions [17].

The extensive body of research produced by this team gives an overview of the issues visually impaired people face in social interactions, and explores many angles for assistive technology solutions to such issues. It has been a major input to the FaceSpeaker project, especially when it came to defining the requirements for the FaceSpeaker prototype. Even though their initial publication concerning a wearable face recognition device dates back to 2005, many of their results and some of their technical information are still remarkably relevant as of 2013.

To the author's knowledge, those 2 projects are the only published academic projects which yielded wearable face recognition devices for the visually impaired. However, other projects proposed wearable face recognition devices

targeting other user groups. A research team from the Japanese Osaka Prefecture University published a paper proposing a covert wearable face recognition device to support human memory [31]. The 3 main requirements they identified are the need for unobtrusiveness, the need for high recognition accuracy and the need for fast recognition. They performed experiments on a desktop computer proving that the required accuracy is attainable within the time constraints imposed by the social interaction context using current technology. They do not provide much details about the actual device's design beyond the need for unobtrusiveness and the fact the name of an identified person would be shown on a head mounted display. In future publications they intend to develop an actual prototype.

Finally, the United States federal governments SBIR (Small Business Innovation Research) program [28] issued a USD 138000 award to the company Advanced Medical Electronics corporation in 2009 for the development of a wearable face recognition device to aid early stage Alzheimer patients [6]. The inability to recognize people is a prominent feature of Alzheimers disease [9]. The proposed device would feature an unobtrusive camera and a Bluetooth wristwatch. The watch would vibrate when the device recognized an acquaintance. It would display the acquaintances name and possibly other relevant information, such as the relationship of the identified person to the user (E.G. neighbor or daughter-in-law). Unfortunately there do not appear to be any publications about this project beyond the extended abstract, and attempts to contact Advanced Medical Electronics Corporation failed.

2.2 "the vOICe": Seeing With Sound

"the vOICe" is a sensory substitution system which converts camera images into soundscapes [23]. Appropriately trained users can interpret those soundscapes and form a mental image of the picture captured by the camera, essentially allowing them to "see with their ears". This project is surrounded by a very active community. "The vOICe"'s setup is based on camera glasses connected to a laptop in the user's backpack, and an Android application is available which can run on Google Glass. While sensory substitution is outside the scope of the FaceSpeaker project, "the vOICe" provided a lot of technical inspiration when designing the prototype.

2.3 Google Glass

As of 2013 various companies are developing augmented reality glasses for sighted users. The best known and most widely publicized example is the Google Glass project [12]. Google glass is a wearable computer worn like a pair of glasses. It runs Google's Android operating system, the same operating system powering many contemporary smartphones and tablet computers. A "heads-up display" mounted above the user's right eye displays information to the user, and audio is conveyed to the user using a bone conduction earpiece. A camera aligned with the user's field of vision is integrated. The device is controlled through speech recognition and a touchpad on the side of the glasses [11].

In principal Google Glass can run any Android application displaying the output on the heads-up display. This setup enables a huge range of applications,

potentially including face recognition applications. It would be possible to display the name of a person on the heads-up display when he comes in view of the camera. Many parties are interested in developing face recognition applications which would run on Google Glass. However, due to privacy concerns Google does not currently permit developers to develop face recognition applications [5].

Google Glass is attracting a lot of media attention and when discussing FaceSpeaker most people immediately compare it to Google Glass. Although Google Glass is not designed for visually impaired users, it could potentially be a platform for running FaceSpeaker software or other computer vision software for the visually impaired. If the use of Google Glass style augmented reality products became socially accepted, this could potentially allow visually impaired users affordable access to computer vision software in a socially acceptable manner. At this time, "the vOICe" (a program which converts camera images into soundscapes discussed in section 2.2) can already run on Google Glass although functionality is still severely limited.

Figure 1: A man controls Google Glass using the touchpad built into the side of the device. Source: wikimedia commons.

While in-depth discussion of augmented reality glasses for sighted users is outside the scope of this paper, some of Google Glass's design elements and social acceptability issues are relevant and will be discussed in subsequent chapters.

2.4 Orcam

In June 2013 the Israeli company Orcam released a commercial computer vision device for the visually impaired. [27] [19]The device consists of camera glasses including a bone conduction earpiece (for covert audio output), hooked up to a processor unit. The device and various concepts surrounding it have been patented [25]. This device can perform various functions such as reading text, telling the user if a traffic light is red or green, and recognizing faces. Some features including face recognition are still in development.

Operating the device requires the user to point at objects. Therefore, the current device is only suitable for users who have enough residual vision to perform such gestures.

3 FaceSpeaker PACT Analysis

" People use technologies to undertake activities in contexts." [4]

In this chapter a PACT analysis for the FaceSpeaker device is performed. This PACT analysis gives an overview of characteristics for each of the 4 PACT elements (People, Activities, Contexts, Technologies) which are relevant to the FaceSpeaker design. More importantly, it "scopes out" some of the variations for different people, activities, contexts and technologies. By making those characteristics and the variations in such characteristics explicit, the PACT analysis highlights important issues which will be used in motivating design choices.

3.1 People

The purpose of FaceSpeaker is to help a user identify people around him, so the most important stakeholders in the FaceSpeaker system are the user and all the people the user interacts with.

FaceSpeaker is primarily designed for visually impaired users. This includes both totally blind users and users with highly variable levels of residual vision. FaceSpeaker must be designed such that users possessing widely varying visual abilities can use it effectively. In certain areas, requirements for blind users fundamentally differ from requirements for partially sighted users and even within the group of partially sighted users requirements will vary according to the level of residual vision.

Users prefer wearable assistive devices to look inconspicuous. If a device "looks weird" such that it is immediately apparent that the user is wearing an assistive device and wearing the device makes the user "stand out" visually, many users would feel uncomfortable using the device. This is especially true in contexts involving social interactions. This was pointed out by many potential FaceSpeaker users, and is also stressed in many related papers [15] [3]. As Krishna et al. put it [15], "Don't make me look like a Martian!" is a sentiment shared by many users when discussing the design of wearable assistive devices.

The FaceSpeaker device is likely to raise privacy concerns. When discussing FaceSpeaker or demonstrating the prototype, people show ambivalent reactions towards the device. On the one hand they are curious about the device and see how it can benefit its users. On the other hand many people report feeling uneasy about a device watching them and covertly telling their name to its user. Almost everybody mentions privacy concerns, notably due to the use of a hidden camera and concerns about the device somehow being linked to internet services. While people are likely to explicitly voice their concerns in the context of a research demonstration, it is more than probable that some people may (consciously or subconsciously) change their social behavior towards an individual using FaceSpeaker, or that they would even avoid this individual.

In addition, when a visually impaired person first meets a sighted person this will often be the first time the sighted person has met a visually impaired individual. This can lead to a lot of questions and (emotional) reactions by itself. If the visually impaired individual is then using a "strange" device, which people have never encountered before and which may engender feelings of unease, the "shock effect" of meeting a visually impaired individual could be amplified. Whether this "shock effect" has a negative impact depends entirely on the at-

titudes and behaviors of the two individuals involved, but it is clearly a factor to keep in mind.

Physically, people may vary significantly in height. This may be an obstacle to getting faces in view of the camera. Other physical obstacles to face recognition could include facial hair or the use of (sun)glasses. At this time FaceSpeaker is an experimental product which would only be used by technically savvy "early adopters". Therefore FaceSpeaker is designed for adult users of normal intelligence who have reasonable experience using assistive technology products including mobile devices.

Visually impaired people may use their auditory sense to compensate for their lack of visual information during social interactions. For example, many visually impaired individuals are adept at identifying people and/or discerning characteristics such as gender and approximate age by listening to people's voices. Auditory cues may help in locating people, knowing what is going on in a room and knowing where such incidents are taking place. High levels of background noise as encountered in many social interaction settings can interfere with hearing (to illustrate: many visually impaired individuals report avoiding situations where loud music is played for this reason). It follows that the user's "auditory attention" is a scarce resource which should be economized on where possible.

3.2 Activities

Essentially, FaceSpeaker supports the user whenever he joins in some activity as part of a relatively large group of collocated individuals. In particular, FaceSpeaker supports activities related to social interactions by helping the user to identify persons. Examples of such activities include "initiating conversation", "finding a person" and "greeting a person". Such activities have highly variable characteristics but in all cases "identifying a person" is a major part of such activities. Essentially, it is this activity of "identifying a person" which FaceSpeaker supports.

Viewed at a high level and ignoring many subtleties, "identifying a person" is a fairly well defined activity. A person perceives some clues (primarily the result of face recognition, possibly gate, hair, clothing etcetera, voice etcetera). Based on those clues and the context, the person decides if the other person is an acquaintance and if so remembers his name in a very short time.

People possessing normal face recognition ability can identify an acquaintance almost instantly and subconsciously. This statement is an oversimplification which ignores many subtleties of human memory and face recognition ability, but is quite close to reality in many situations. For example, most persons will immediately pick out a close friend in a large group of people.

In summary, the activity of "identifying a person" is an extremely frequent activity which happens in a huge variety of contexts and is typically carried out almost subconsciously. This implies that any technology designed to support this activity should be designed for high responsiveness and very efficient operation.

"Identifying a person" is not a safety critical activity for a sensible adult involved in ordinary social interactions. However, failure to recognize an acquaintance or incorrectly recognizing an acquaintance can cause socially embarrassing situations or lead to lost social opportunities in fact that is the very reason for

developing FaceSpeaker. It should be noted, however, that normal face recognition abilities do not guarantee success in this activity either. It is normal for people to misidentify a person or forget a name on occasion especially if they meet acquaintances they have not met in a long time.

3.3 Contexts

Typically, FaceSpeaker will be used in contexts where a relatively large group of persons is collocated and where the user is heavily engaged in social interactions. Those social interactions will occupy much of the user's attention. In addition, focusing attention on devices is often socially unacceptable think of the times you tried to keep a conversation going while your conversation partner was writing messages on a smartphone. Overt audio output is not acceptable in such contexts, because it can disrupt conversation and focus unwanted attention on the user.

FaceSpeaker will often be used in contexts involving high levels of background noise. This is a major problem for both audio input and output. High levels of background noise can prevent the user from hearing audio output, and technologies such as speech recognition are typically useless in the presence of high background noise levels.

Finally FaceSpeaker will be used under highly variable lighting conditions. Variations in lighting conditions can have a major impact on both cameras and face recognition algorithms.

3.4 Technologies

FaceSpeaker is a wearable system intended for intensive mobile use. Therefore the hardware should be small, lightweight and durable. For the user's comfort, any heat production should be kept to a minimum and battery life should be maximized. This is challenging mainly due to the high computational demands of live face recognition algorithms.

The device needs to be unobtrusive. This poses major technical challenges. A key challenge is finding a suitable body mounted camera. On the one hand, the camera needs to be very small and unobtrusive. At the same time, face recognition algorithms require high quality images and the camera needs to work well under highly variable conditions. Another issue is unobtrusive audio input and output which will require specialized technology.

4 General Requirements: Unobtrusiveness and User Attention

Based on the literature study and the above PACT analysis, it is concluded that the 2 most important high level requirements for the FaceSpeaker design are unobtrusiveness and the need to economize on the user's attention.

All cited projects which have proposed a wearable face recognition device explicitly identify the unobtrusiveness requirement. FaceSpeaker should be unobtrusive in both appearance and operation. As mentioned in the PACT analysis, wearable assistive devices should not make people "stand out" if they are to be accepted by users. This also follows from the ICare interaction assistant team's extensive user study and systematic requirements analysis [15].

Beyond this argument, the requirement that the user should be able to unobtrusively control FaceSpeaker independently arises from the social interaction context in which the device will be used. In many social interactions, such as a simple one on one conversation, social norms mandate that people focus near-exclusive attention on the social interaction. If the user is perceived to shift his attention to another activity, such as controlling FaceSpeaker, this can easily be perceived as a lack of interest or even rudeness.

This argument also implies the requirement to economize on user attention. In addition, this requirement follows from the context in which many social interactions take place, where high levels of activity and background noise are commonplace and many stimuli compete for the user's attention. The ICare interaction assistant team identified the risk of "auditory overload", which was part of their motivation to develop a haptic belt [20]. However, none of the cited studies explicitly take the issue of effectively engaging the user's attention into account. This issue will be taken into account when discussing the FaceSpeaker design in subsequent chapters, and it is suggested that future studies pay more attention to the challenging issue of properly handling user attention in a social interaction context.

5 The Prototype

This chapter first describes the FaceSpeaker prototype. It then gives a high level overview of the development process and the technology involved. Appendix A provides details of the research performed to select a camera for the FaceSpeaker prototype, while appendix B provides details on multithreading and performance optimization of the FaceSpeaker software. Appendix C contains the user guide for the FaceSpeaker software.

Technology was not the focus of this study. Therefore this chapter discusses the technology involved on a very high level and leaves out most technical details. Readers interested in such details may consult the cited references, and the full FaceSpeaker source code is available at the facespeaker.org website for programmers who are interested in FaceSpeaker's inner workings.

5.1 Overview

The purpose of the FaceSpeaker prototype is to automatically identify the users acquaintances when they come in view of the camera. The user wears Lawmate analogue camera glasses and a backpack. The backpack contains a laptop and the "camera box" which holds the hardware for powering the camera glasses and sending the video stream to the laptop (see section 5.2.1 for details). The user controls the laptop using a small Bluetooth keyboard (see section 5.3 for details). Sound output is provided using bone conduction headphones which do not obstruct the user's ears.

Figure 2: FaceSpeaker author wearing the FaceSpeaker prototype.

The laptop runs Microsoft Windows 7 professional X64 edition and the open source FaceSpeaker software. See the FaceSpeaker user manual (appendix C) for an extensive description of the FaceSpeaker software's functionality. A comprehensive video demonstration is also available on the FaceSpeaker.org website.

Figure 3: The FaceSpeaker user interface.

In summary, the software can be trained to recognize the users acquaintances. The user can add an acquaintance to FaceSpeakers database by typing that persons name on the small keyboard and pressing enter to start the training procedure. The user then needs to look in the direction of the persons face so the camera can capture 20 training images. The software confirms the camera is pointed correctly by issuing a click sound whenever a training image is captured.

During normal operation, the FaceSpeaker software constantly monitors the video stream captured by the camera and reacts whenever a face comes in view of the camera. If the detected face is unknown, the program issues a single low pitched beep. If the face belongs to a person enrolled in the database, the software issues a high pitched beep (the higher pitched the beep, the more confident the identification) and speaks the persons name about half a second later. The beep mainly serves to shift the users attention towards the verbal utterance of the identified persons name.

The FaceSpeaker software also enables the user to manage the database of enrolled persons. A list of enrolled persons is provided. It is possible to add training data for an enrolled person (in order to improve recognition accuracy), existing training images may be visually inspected and persons may be removed from the database.

5.2 The Camera

After an extensive investigation considering various camera positions and options, the Lawmate cm-sg10 camera glasses were selected for the FaceSpeaker prototype. For a detailed motivation and details about the cameras which have been investigated, read appendix A. The Lawmate glasses feature a practically invisible camera in the right arm and a microphone in the left arm. The glasses have a professional design and can be fitted with sunglass, transparent or prescription lenses. The possibility to use transparent lenses is critical because sunglasses (as used by "the vOICe" and some related prototypes) are not socially acceptable in many indoor settings and can interfere with the user's vision.

Figure 4: FaceSpeaker author wearing the Lawmate camera glasses fitted with sunglass lenses.

5.2.1 Camera Box

Hooking the Lawmate camera glasses up to the laptop proved challenging. The camera glasses are usually connected to a specialized mobile video recorder using a single plug. Fortunately, the camera glasses include a conversion cable for hooking the glasses up to standard video equipment featuring an RCA S-video input. When using this cable, a 9 volt power supply is required to power the camera glasses. A battery holder is included to power the glasses using a 9 volt battery.

Figure 5: The FaceSpeaker camera box. On one side there is a single USB cable which is connected to the laptop. The camera glasses are connected to a cable coming out at the other side.

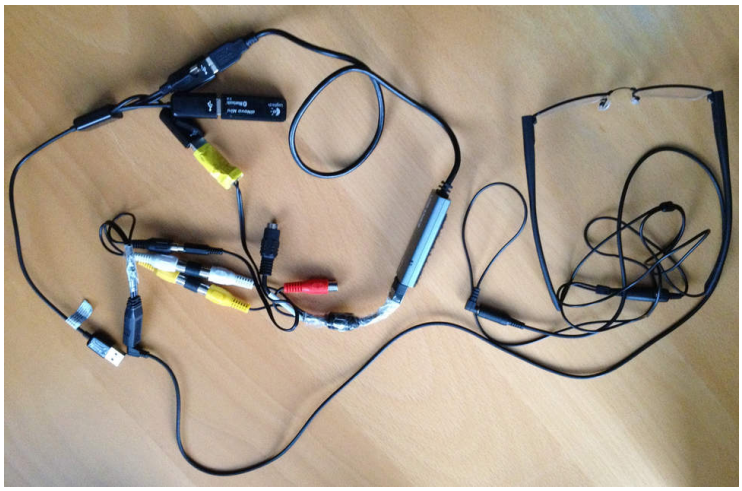

Figure 6: An overview of the setup used to power the camera glasses and connect them to the laptop. The camera glasses are connected to the conversion cable. From the conversion cable, RCA video and audio plugs are connected to the Hauppauge video capture device, and the power input plug on the conversion cable is connected to the USB voltage booster. The video capture device and the USB voltage booster are both connected to a USB hub, to which is also connected a receiver for a Bluetooth keyboard.

The first challenge was to convert the analogue video signal into a digital signal. Many devices for grabbing analogue video signals are commercially available. The most common use for such devices is digitization of old video tapes. Unfortunately, most such devices can only be used with the supplied proprietary video capturing software. After unsuccessfully trying various devices, the Hauppauge USB Live-2 video capture device was selected. This device behaves like a standard USB webcam and allows most software including FaceSpeaker to access the video signal.

Powering the camera glasses using a 9 volt battery was not practical. The battery would not last more than a few hours and it was quite a hassle to connect and disconnect the battery before every use of the prototype. It seemed desirable to power the camera glasses from the laptop's USB port. However, the USB port supplies 5 volts of power while the camera glasses require a 9 volt power supply. "the vOICe" suggests a solution to this problem on their website using a USB voltage booster [24]. The USB voltage booster shown in the image was ordered from Ebay. This device can be plugged into a standard USB port and boosts the voltage from 5 volts to 9 volts. A power plug compatible with the camera glasses was connected to the power output terminals such that voltage and polarity were

Figure 7: The 5W USB DC-DC 5V to 9V Step Up Boost Module. This voltage booster can be plugged into a standard USB port and outputs 9 volts of power from 2 terminals into which wires can be screwed.

identical to the voltage and polarity supplied
by the battery holder.

The video grabber, voltage booster, a USB wire hub and all the cabling were combined into a cigar box and fixed in place using adhesive Velcro. The result is a "camera box", which makes it possible to get the prototype ready for use by connecting just 2 cables (the camera glasses and a single USB plug to the laptop). This setup also prevents a potentially hazardous mess of cabling in the backpack.

5.3 Controlling The Laptop

The user controls the laptop using a small Bluetooth keyboard. Users proficient in the use of screen readers and windows keyboard navigation can access all the laptops functionality using this keyboard. The selected Riitek mini Bluetooth keyboard is approximately the size of a smartphone. Because the keyboard has excellent tactile characteristics, most blind users can learn to operate it efficiently. Because some laptops cannot establish a reliable Bluetooth connection with the keyboard when the laptop is in a backpack, it may be necessary to use a separate USB Bluetooth adapter which can be placed in the camera box.

Figure 8: Riitek mini bluetooth keyboard used to control the FaceSpeaker prototype.

While the keyboard performed well when field testing the prototype, this cannot be the only means to control a practicable device. As demonstrated in chapter 7 on enrollment, the keyboard is too obtrusive and using it places too much demands on the user's attention in a social interaction context. Some related devices including Astler et al's prototype [3] and "the vOICe" [23] rely on speech recognition technology for operating the device. However, speech recognition is not an appropriate method for controlling FaceSpeaker. It is too obtrusive, and "talking to a computer" is not socially acceptable in most social interaction contexts. The high levels of background noise encountered in such contexts will often prevent the use of speech recognition technology, and issuing voice commands may be too time consuming (consult section 8.1 for more information).

As will be demonstrated in chapters 7 on enrollment and 8 on person identification, there should be a few buttons which the user can activate quickly and unobtrusively. The FaceSpeaker project did not investigate how such buttons should be placed, but options may include placing some buttons on a wristwatch(inspired by Advanced Medical Electronics Corporation's proposed face recognition device for Alzheimer patients [6]) or designing a small control box which could be worn in a trouser pocket. Such a simple control mechanism would have to be supplemented with speech recognition technology or a small keyboard for performing more complicated tasks such as managing the database

of enrolled persons. Designing a control mechanism which is sufficiently unobtrusive for use in a social interaction context is challenging and suggested as a topic for future research.

5.4 Audio Output

The main requirements when issuing audio feedback to the user in a social interaction context are that the user's hearing is not obstructed while people around the user do not hear the audible output. This problem is commonly solved using bone conduction technology. Examples of related devices using such technology are Google Glass [12] and Orcam's device [27].

Figure 9: A man wearing the AfterShokz bone conduction headphones.

A bone conduction earpiece is placed just behind the ear as shown in the below image. The earpiece vibrates such that the vibrations are conducted by the scull to the wearer's inner ear. This means the audio will blend in with the sound the user hears through his ears while it is barely noticeable to bystanders. In an actual device a bone conduction earpiece would be unobtrusively mounted to an arm of the camera glasses as in the Google Glass and Orcam devices. For the FaceSpeaker prototype, a commercially available bone conduction headphone such as the AfterShokz bone conduction headphones shown in figure 9 [1] can be used. Such headphones are already successfully used by blind users to listen to speech output of mobile devices in public.

5.5 FaceSpeaker Software Development

Note: the remainder of this chapter describes technical details of the FaceSpeaker software and face recognition algorithms. The reader is assumed to have some technical background. Readers not interested in technical details of the FaceSpeaker software may proceed to chapter 6.

The FaceSpeaker software runs on X64 editions of Microsoft Windows 7 or Microsoft Windows 8. It was developed in the Microsoft Visual C# programming language using the Microsoft visual studio 2010 ultimate edition integrated development environment.

Targeting a smartphone or mobile device platform was considered but dismissed as infeasible. Live face recognition is a computationally intensive task. To illustrate, the Luxand face SDK manual [13] (see section 5.6 for more information) recommends using a "intel core I7 or Xeon" processor. While mobile devices feature increasingly powerful processors, as of early 2013 no mobile devices come close to providing computational power comparable to an advanced desktop processor. So even if a mobile device could perform live face recognition, this would require highly optimized code. The FaceSpeaker project did not have the development time available to make such optimizations. In addition,

it was very difficult if not impossible to connect camera glasses to any mobile device. Even if this had been possible, targeting a mobile platform would have introduced technical complications and would have complicated the development process. The FaceSpeaker project did not have the development time or knowledge available to enable this extra investment.

The advantage of targeting the .net platform is that it is native to Windows. This is likely to yield the best user experience, best screen reader accessibility, highest performance and most stable hardware interaction achievable on the Windows platform, while requiring the least amount of development effort. The C# programming language was chosen because it is a modern, advanced language which is easy to use and particularly suitable for developing multithreaded applications. FaceSpeaker was developed as a native 64 bit (X64) application because this may provide performance and stability advantages. Since FaceSpeaker only works well on computers featuring a quad core processor and at least 4GB of random access memory, it does not normally make sense to run FaceSpeaker on an older computer running a 32 bit (X86) edition of Windows. However, it is possible to compile FaceSpeaker as a 32 bit application if desired.

Because live face recognition is a computationally intensive task, FaceSpeaker must take advantage of all the processing power available. Because modern computers feature multicore processors, this requires writing a multithreaded application. Much of the development time was spent ensuring FaceSpeaker became a reliable multithreaded application. Compared to other programming languages, the C# programming language features advanced native functionality to facilitate easy development of reliable multithreaded applications.

A typical camera can capture about 30 images ("frames") per second. Most laptops cannot process frames at that rate. Therefore FaceSpeaker automatically adjusts the camera's frame rate to the processing power the computer has available. Higher frame rates result in more robust face recognition, but put a high strain on the laptop's processor. High processor utilization results in potentially uncomfortable or hazardous levels of heat production and dramatically shortens the laptop's battery life. Therefore a power saver function was implemented which decreases the frame rate to 4 frames per second when no face is in view of the camera or the same face stays in view of the camera after it has been identified. When a face is detected the frame rate is immediately restored to the maximum rate attainable by the laptop. Lowering the frame rate results in a dramatically lowered processor utilization with a corresponding drop in power consumption and heat production. The power saver proved effective during field trials. The laptop did not run excessively hot and battery life was acceptable while FaceSpeaker performance was not significantly affected when the power saver function was enabled.

The FaceSpeaker source code has a modular design. Most software components including the user interface and "face engine" (the code responsible for face detection and recognition) are programmed against interfaces (IUserInterface, IFaceEngine etcetera). This makes tasks such as using a different face recognition library or developing a different user interface particularly easy.

For more elaborate technical information about FaceSpeaker's operation, multithreading and performance optimization, consult appendix B. Programmers interested in FaceSpeaker's software design may download the full source code from the facespeaker.org website.

Finally, an effort was made to make the FaceSpeaker user interface accessible to screen reader users. Every component was properly described using the .net framework's accessibility functionality. Every component was assigned an access key, so the user can move focus to any component by pressing a single hotkey. The user interface features a "self voicing" option, so blind users can disable their screen reader while using FaceSpeaker (this may result in better performance). FaceSpeaker uses the Microsoft Speech API version 5 ("sapi5") for speech output. The user can use the standard Microsoft speech synthesizer included with Windows or use any other sapi5 compatible speech synthesizer for better quality and responsiveness.

5.6 Face Recognition Algorithms

This section discusses the face detection and recognition algorithms used by FaceSpeaker.

5.6.1 The Face Recognition Library

The first step was to choose a face recognition library. Various commercial face recognition libraries are available. The commercial option which has been considered was the Luxand faceSDK version 4.0 [13]. This is a well-documented library which can be downloaded for free time-limited evaluation. Astler et al. used this library for building their prototype, and report that the face recognition algorithms worked reliably [3]. In September 2013 Luxand released version 5.0 of their faceSDK, which is claimed to have much better recognition accuracy and some other new features. Notably, the new version includes features which would make the process of enrolling a person much faster and which automatically adds training data when the same person is encountered in different lighting conditions. The FaceSpeaker development time frame had closed at the time this version was released, but the new functionality could significantly enhance future prototypes.

The FaceSpeaker project did not use Luxand's product because this would have prevented the FaceSpeaker software from being released as an open source product. In addition, purchasing a commercial product was not feasible due to FaceSpeaker's minimal budget. Ultimately the goal of developing the FaceSpeaker prototype was not to get a perfect face recognition device - that is an unrealistic goal for a small project carried out by a single student. Rather, the goal was to get a proof of concept which would benefit the academic community and work well enough for simple field trials which could inspire design recommendations. Therefore it was decided to attempt using open source libraries even if this resulted in suboptimal performance. However, the FaceSpeaker software has a highly modular design and the face recognition algorithms are loosely coupled to the rest of the program. This makes it easy for programmers to experiment with other (commercial) face recognition algorithms by implementing the IFaceEngine interface and some related interfaces.

The choice for an open source library having been made, the next question was which library to use. The de facto standard for open source face recognition products is the openCV library [26]. This library features many computer vision algorithms including face detection and face recognition algorithms. While some other open source algorithms for face detection and recognition have been

found on the internet, none of the code fragments available were sufficiently documented or easy to implement for the purposes of quickly developing a working prototype and there was no evidence that any such solutions would outperform openCV. OpenCV has a wide user base and is under very active development, so it was decided to adopt this de facto standard. OpenCV is a native C++ library, but the well documented and actively maintained emguCV wrapper for using it in C# is available [7].

5.6.2 Face Recognition Algorithm Selection And Optimization

The starting point for the FaceSpeaker software was Jonson's live face detection and recognition C# application available on CodeProject [14]. The reader interested in technical details of emguCV and the face recognition algorithms used is referred to Johnson's article as well as the openCV and emguCV documentation available on the websites of those libraries [26] [7]. Programmers interested in the details of how FaceSpeaker implements those algorithms may consult FaceSpeaker's source code available on the facespeaker.org website.

EmguCV version 2.4.2 implements 3 face recognition algorithms: the Eigenface recognizer, the Fisherface recognizer and the LBPH face recognizer. All 3 types of recognizers can be trained and used in a similar manner as explained in Johnson's article [14]. Experimentation was performed to find the face recognition algorithm which was most reliable under variable lighting conditions as encountered while using FaceSpeaker. This investigation was extensive but informal. Therefore, the below discussion only gives a high level overview of the qualitative results.

The Eigenface recognizer was useless in practice. No matter how it was configured, it would either fail to recognize enrolled persons even under the most trivial changes in lighting conditions or make such absurd misidentifications that it performed at or below the level of a random guess. For the Fisherface and LBPH recognizers, settings could be found which enabled the recognizers to tell people apart under variable lighting conditions.

While experimenting with both recognizers, two observations were made. First, it was apparent that both recognizers were resilient to different types of lighting changes (e.g. if a person was enrolled and the camera was moved, usually one of the algorithms would still recognize the person while the other algorithm would not). It was also observed that if an algorithm made a correct identification, it would usually make the same identification consistently across different frames where as a recognizer which was making a wrong identification would constantly "change its judgment" across different frames. This lead to the idea that combining the 2 face recognition algorithms and developing a scoring system could result in a better chance of making a successful identification. The idea of combining multiple face recognition algorithms to increase robustness under variable lighting conditions was conceived and successfully applied in earlier studies [18].

5.6.3 The FaceSpeaker Pipeline

The manner in which FaceSpeaker combines frames captured by the camera to identify a person can be roughly summarized as follows.

- The camera constantly captures images ("frames").
- When the camera has captured a frame, emguCV's face detection algorithm is run to find a face on the image. If no face is found the frame is discarded, if a face is found the rectangle containing the face is "cut out" of the image and passed on to the Fisherface and Eigenface recognition algorithms.
- Both algorithms identify the face. They either report that the face is unknown (it belongs to a person who is not enrolled into FaceSpeaker's database), or they report the name of the enrolled person the face belongs to. In the latter case, they also return a measure of "distance" which gives an indication of the likelihood that the identification is correct.
- In order to identify a person, FaceSpeaker iterates over all the frames captured and processed within the last 2 seconds. For every candidate in this list the procedure keeps a score which is increased every time the candidate is identified on a frame. Some of the heuristics used when calculating the contribution an identification of a candidate on a given frame makes to the candidate's overall score include:
  - The newer the frame, the higher the contribution. This serves to improve FaceSpeaker's responsiveness in case the camera's view changes.
  - In case of an identification other than "unknown", the contribution depends on the "distance" returned by the face recognition algorithms (which is a measure for the confidence of that identification).
  - If on a given frame both recognition algorithms make the same identification, the contribution is significantly increased because the identification is very likely to be correct if both algorithms agree on it.
  - If one (or both) algorithms consistently make the same identification accross frames, this increases the contribution.
- The candidate who has the highest overall score is returned as the identified person ("unknown" is considered as a candidate in this competition).

In reality the frames are being processed in parallel which introduces some complications mainly related to timing. More details can be found in appendix B. Programmers interested in full technical details of the FaceSpeaker pipeline may download the FaceSpeaker source code from the facespeaker.org website.

5.6.4 Adding Training Images To Enhance Recognition Accuracy

As noted in the previous section, enrolling a person involves adding 20 training images to the database. The idea is that the user will move about a bit and capture the face in slightly different poses and under slightly different lighting conditions. Even with the LBPH and Fisherface algorithms combined as described above, a person would not usually be recognized if the prototype was

moved to a different room. Therefore a function to add training data for an enrolled person was implemented. If the user notices that a person is not recognized in a new situation, he can select that person's name in the "trained person" list and activate the "add training data" button to start this procedure. The procedure adds a maximum of 20 additional training images to the database, but only adds an image if it is not properly identified by at least one of the face algorithms. This ensures that the retraining procedure only adds images which will actually increase the chance of successful identifications in practice.

Figure 10: The same face captured in different locations. This figure illustrates the dramatic effects of changes in elumination, camera movement and other factors.

5.6.5 Results

The FaceSpeaker prototype performs reasonably under laboratory conditions. If the camera is placed at a fixed position and various people are enrolled into FaceSpeaker, FaceSpeaker will tell those people apart fairly reliably even if they move about or lighting conditions slightly change (this is a vast improvement over results achieved using the Eigenface recognizer or even the unenhanced LBPH or Fisherface recognizers). Unfortunately, the prototype does not work reliably in mobile use. If a person is enrolled he will initially be recognized, but if the same person is later encountered in a different room identification will usually fail and retraining is necessary. Only after various retraining sessions in different rooms will the prototype begin to reliably identify a person which makes it unsuitable for use outside a demonstration setting. The problem appears to be that lighting conditions, camera movements and other factors have a dramatic impact on the captured face images. Figure 10 illustrates this problem. OpenCV features the "histogram equalization" algorithm which is supposed to compensate for variations in lighting conditions, but this is not nearly sufficient to make the face recognition algorithms work reliably given the extreme variations encountered in mobile use. Part of the problem could also be the camera's quality or other characteristics.

The conclusion is that openCV does not currently provide algorithms which are robust enough for use in a practicable FaceSpeaker device, and that a prototype intended for more realistic and elaborate field trials would need to use commercial face recognition solutions at this time. This is the same conclusion which Astler et al. drew when they compared open source solutions to commercial solutions [3]. The good news is that openCV is under very active development and that the situation seems to be considerably improved compared to the situation at the time of Astler et al's study. OpenCV now provides various well documented face recognition algorithms, and thanks to the enhancements described in this section the FaceSpeaker prototype did perform well enough for fairly realistic demonstrations and limited field trials. The FaceSpeaker author is optimistic that with some more development open source face recognition algorithms will reach the necessary level of robustness, and hopes that the FaceSpeaker project will help openCV developers to gain a better understanding of the problems involved, and that the enhancements described in this section may serve as a starting point for the openCV community to improve their algorithms.

6 The Small Conference Scenario

Due to the extreme complexity, subjective codes and subconscious behavior involved in human social interactions it is impossible to objectively define what impact FaceSpeaker-like technology will have on a given social situation. It is, however, possible to gain an appreciation for the issues involved and make motivated design decisions based on that intuition.

The small conference scenario illustrates how FaceSpeaker could be used in an actual contexts and shows examples of how the user and people around the user might react to this technology. This serves to give the reader an appreciation of the issues involved, and provides a basis for discussions about the FaceSpeaker design.

The scenario is a fictional account of a FaceSpeaker user attending a small academic conference. The context of a small academic conference was chosen because many of the situations in which FaceSpeaker could be beneficial occur, the social context is relatively easy to define and this type of setting is widely familiar in the academic community. Some contextual elements are similar to the situation at ICCHP Summer University 2013 in Karlsruhe [30], the event where the prototype was mainly tested. The scenario is completely fictional, but was inspired by many comments from discussions with potential stakeholders and experience gained while field testing the prototype.

The scenario grossly oversimplifies and stereotypes human behavior and thought processes, such that many of the events are somewhat unlikely in practice. It leaves out many subtleties, such as the likelihood that conference participants would wear name badges and hotel staff members might be in uniform. Oversimplification is necessary to keep the scenario concise and readable. But it is mainly a deliberate choice to encourage brainstorming and discussions. Presenting extremes and stereotypes serves to elicit reactions. Coming up with nuances and subtleties is left to the reader. It is precisely by performing this task, and by discussing such nuances and subtleties with other people, that some feeling and insight can be gained into the very complicated matter of how a FaceSpeaker device would impact social interactions in practice.

For reasons of linguistic simplicity and unambiguity when referring to people, the user will be referred to in the first person throughout the scenario and design discussions in subsequent chapters. When writing this type of scenario, the author's personality and perspectives inevitably exert a large influence. While an effort is made to highlight different user perspectives and keep the user a neutral figure, it is important to realize that the user in this scenario is ultimately modelled on the FaceSpeaker author. He is a 25 year old male student who has an extremely limited residual vision. At close range he can see where people are standing and discern a few characteristics such as hair color, clothing or gender. However, he does not have any face recognition ability.

Scenario introduction

The context of this scenario is a 5 day international academic conference with about 50 participants. I will wear the FaceSpeaker device to help me recognize people around me. I do not know most of the participants, and most of the participants do not know each other (hence it is acceptable that people explicitly introduce themselves). The conference takes place in a hotel. There is a coffee area, bar / restaurant and there are some workshop rooms. All other participants are normally sighted. When I arrive at the hotel my FaceSpeaker

device is new and empty.

In the coming chapters, the requirements and design choices for various aspects of the FaceSpeaker design are discussed. The scenario is "waved through" the remainder of this paper: at various points sub scenarios will be mixed in with the text to motivate requirements and illustrate points made in the text.

7 The Enrollment Procedure

Before FaceSpeaker can identify a person, that person must be added to FaceSpeaker's database ("enrolled"). For every person, FaceSpeaker needs a name and a set of training images. The enrollment procedure is fairly simple: type a name on the small keyboard, press enter and point the camera at the person for a few seconds. In case of a cooperative, understanding subject it can be very easy as described in the below scenario.

Scenario 1:Enrolling an old friend

This scenario describes the prototype's enrollment procedure, and serves to illustrate how person enrollment would ideally be carried out.

As I enter the lobby I am greeted by Jonathan, one of the participants I did meet on previous occasions. We talked extensively then, he is aware of my disability and the assistive technology I use. So we chat some and I tell him about my cool new FaceSpeaker device. He thinks the device is very interesting and I show him how it works.

I look at his face. I hear a low pitched beep, indicating his face is now in view of the camera. I take the small keyboard, type "Jonathan" and hit enter. Now I hear FaceSpeaker say "start training, ensure the person to be added is looking at the camera". I look in the general direction of his face (which is fairly easy as we are about the same height and standing by ourselves facing each other). I have to keep looking at his face for a few seconds so FaceSpeaker can capture enough training images. After a few seconds I hear a sound indicating the enrollment procedure is finished. When I turn my head, I hear a long low pitched beep indicating Jonathan's face is no longer in view of the camera. When I turn my head back, I hear a high pitched beep and FaceSpeaker says "Jonathan". So it works!

7.1 Covert Enrollment and Grouping

The above "ideal" scenario assumes Jonathan is an acquaintance who is aware of my disability and has a positive attitude towards the FaceSpeaker device. It assumes that the circumstances are such that we have time for an extended conversation, during which it would be possible and appropriate for me to explain FaceSpeaker and ask for permission to enroll Jonathan. Nearly all of the people I meet at the conference will be new acquaintances. Often introductions are fairly brief affairs which leave no time or appropriate opportunity to introduce FaceSpeaker and explicitly carry out the enrollment procedure. In some introductions it is not even appropriate or necessary to exchange names, but I may want to get a clue that I met a person before or that a person belongs to a certain group. This point is illustrated by the below scenario.

Scenario 2: Enrolling the receptionist?

This scenario illustrates a situation in which overtly enrolling a person is not feasible. It demonstrates the merits of covertly enrolling a person in this situation, and demonstrates the need to divide enrolled persons into groups.

I move to the reception desk to get my room key. When I look in the direction of the receptionist, I hear a low pitched beep; FaceSpeaker has detected an unknown face. Should I enroll her? It would be useful to know that she is a hotel staff member if she approached me or walked by. But now is not the time to make personal introductions. She just has to get me my key and the

people queued up behind me might not appreciate me disturbing her with social Smalltalk. I could be conceived as inappropriately personal if I bluntly asked for her name. I don't really need to know her name anyway, what matters most is that I know she is a hotel staff member. There are so many staff members, constantly hearing their names might just confuse me. On the other hand, if she were around my age and I liked her "at first glance" or based on our brief interaction it happens I might have very good reason to approach this particular staff member if I got an appropriate opportunity. But even then, I don't need to know her name at this time. I just need to know she is that receptionist who gave me my room key. That brings me to the solution. As she turns to search the key, I type "receptionist1" on the small keyboard and hit enter to start the enrollment procedure. When she turns back and her face comes in view of the camera, the clicks start sounding to indicate face images are being captured and after a few seconds FaceSpeaker indicates enrollment is complete. She never noticed and if I met her again I have the clue I need.

In this scenario the enrolment procedure was "covered up". By covering up the training procedure, I avoid introducing FaceSpeaker and focusing (extra) attention on my disability. The downside is that this behavior can be perceived as sneaky. The below scenario illustrates another reason why I might want to cover up an enrollment procedure, and illustrates what might happen if I am caught doing so.

Scenario 3: "Got you!"

This scenario illustrates that people could react very negatively to a covert enrollment procedure.

Somehow I find myself in a conversation with an elderly professor. We discuss some academic matters, but for some reason I don't like her much. I might still want to identify her later if only to avoid getting involved with her again, it is that bad. I did learn her name, and type it on the keyboard. I try to be discrete but she notices and harshly asks: "What are you doing? Is this any time to type on your smartphone?". Wanting to be open, I answer "No, eh I have this device" and tell her something about the FaceSpeaker device and how it helps me. Then she explodes. "How dare you spy on me? I don't care if you are disabled or not, you must respect other people's privacy. You sneaky idiot I will report you to the conference organizers! Get off!". Fortunately she didn't do any such thing, and afterwards I'm having a good laugh with a couple of guys who witnessed the incident. I'm not unhappy to learn she leaves the conference early.

It is more than likely that people who become aware I cover up the use of a hidden camera feel uneasy. They may share their concerns, or they may keep their thoughts to themselves and avoid me. Even if covert enrollment is sometimes necessary, it seems best to be open about the device wherever possible. As noted in the PACT analysis, most people react positively to the device once they understand how it works and how it benefits its users. Openly answering questions and actively informing people will usually result in positive reactions, but covering up the device especially if deception tactics were used will make people feel the user is spying on them and can result in some very negative reactions and social exclusion.

So if I meet a new acquaintance and we start a conversation, I will have to decide whether or not to tell him about FaceSpeaker, and if so, at what point in the conversation I will do so. This is closely related to the issue of telling a

new acquaintance about my disability. This is a controversial and complicated psychological issue. Every disabled person has his own means to deal with this issue, depending on personality and the disability involved. In case of a totally blind person the disability is immediately apparent (and conversation partners will often start asking questions about it fairly soon), while a milder visual impairment is easier to mask and might only be revealed if the conversation reaches a fairly personal level. Such issues are extremely personal and not directly relevant to the enrollment procedure design, but the below scenario illustrates a few of the issues involved.

Scenario 4: Enrolling the insecure guy

This scenario illustrates some issues which may arise when overtly enrolling people and telling them about the FaceSpeaker device. I enter the coffee area. Most people seem deeply engaged in conversation. I notice a man standing by himself who seems approachable, and when I look in his direction I hear a low pitched beep indicating FaceSpeaker detected an unknown face. I try to start a conversation. I learn he is involved in a project I find interesting, but the conversation doesn't go that well. His English seems quite poor and he acts a bit reserved. Why? This is his first international event, so he might feel ill at ease in this environment. He might be a bit shy by personality. Another possibility is that I am the first visually impaired individual this man has ever met. I know that certain aspects such as the fact I cannot make normal eye contact will cause some people to feel ill at ease talking to me, at least initially. Typically such people mainly have a lot of unanswered questions and are hesitant to bring the topic up.

I decide to use FaceSpeaker as an "icebreaker". I ask for his name. I explain that because I am partially sighted, I have problems identifying people and I have a new device helping me with this task. He agrees to be enrolled, and I carry out the enrollment procedure.

Now 2 things might happen. If the issue was that he was a bit "shocked" by the fact I did not make eye contact and he was hesitant to ask questions, me bringing up the issue might help him overcome his hesitations and ask the questions he has on his mind. It could be the conversation gets flowing after that and we become friends for the rest of the conference.

On the other hand, he could stay reserved or even become more reserved. It is possible that the explanation about my disability and/or FaceSpeaker made him feel more insecure than he already felt. Or the fact that we don't get a conversation going may have little to do with my disability. Maybe his English is just too poor, he feels so insecure he wouldn't talk to anyone or our personalities simply don't match.

Overall, it is concluded that the overt enrollment procedure as used in the FaceSpeaker prototype and related devices does not suffice in practice. Asking every person who needs to be enrolled for permission, no matter how desirable that might be, is totally infeasible in all but the simplest social settings. Overt actions, such as typing a name on a keyboard let alone issuing voice commands, inevitably shift attention towards the device violating the unobtrusiveness requirement. Beyond that, FaceSpeaker field trials clearly demonstrated that complex actions like typing a name on a keyboard are too distracting for the user. With practice and a technically improved device it might be possible to make such actions less distracting, but it seems unlikely that such actions would ever become automated enough to be carried out unobtrusively by the average

user in a social interaction context. We have also seen that simply attaching a name to a user does not suffice. Sometimes it is necessary to enroll people whose name is unknown, and it might be useful to divide (anonymous or named) people into groups such as hotel staff or organizers.

7.2 The Proposed Enrollment Procedure

Those conclusions lead to a proposed covert enrollment procedure, which could either replace or complement overt procedures. When the user wants to enroll a person, he gets that person's face in view of the camera and presses a covert button (or performs some other quick and unobtrusive action). FaceSpeaker will then enroll that person and record about 10 seconds of audio using the microphone in the user's camera glasses. People enrolled this way are placed on a "to do" list. When the user has a moment by himself, he can provide the names for recently enrolled people on this to do list and/or assign those people to groups. For every person on the to do list, the time of enrollment and the 10 second audio recording are available to the user. The user should ensure he can deduce the identity of that person from the audio fragment (so typically the point at which people exchange names is the right time for covert enrollment).

Scenario 5: greeting and referencing

This scenario illustrates how the proposed training procedure would work and how it would benefit the user.

On the second day of the conference, I leave my hotel room to head for the plenary opening in the workshop room. In the elevator I see a man. When I look at his face, FaceSpeaker issues a high pitched beep and says "Anthony". I remember his name, I met him in the bar yesterday and enrolled him. We had an interesting conversation. Thanks to FaceSpeaker's feedback I can now say "Good morning, Anthony! How are you doing?". Without FaceSpeaker I either would not recognize him or I would be too much in doubt about his identity to take the initiative. If he were aware of my visual impairment, he might have taken the initiative. But if we were not aware, he might be offended by the lack of recognition and my failure to greet him.

We walk into the coffee area and chat some more on the way. When we enter the coffee area Anthony spots Gerhard, a colleague from his university who is involved in research I would probably be interested in. Anthony introduces me to Gerhard. Just before we shake hands and exchange names, I press the covert enrollment button. I'm glad I did, because 20 seconds later we are called into the workshop room and I lose Gerhard in the crowd. During the boring opening speech, I have an opportunity to pull out my keyboard and find Gerhard on my to do list. If there had been multiple persons on the list, I could have told them apart based on the audio recordings made for each of them. I enter Gerhard's name FaceSpeaker helps me to find Gerhard. We talk extensively, and at the time we get to discuss my disability the FaceSpeaker device is one of the matters I tell him about.

7.3 Conclusion

Requirements for the enrollment procedure are summarized as follows. The device should feature both overt and covert enrollment procedures. In the overt procedure the user enters the person's name immediately. The user must have an

extremely unobtrusive way (covert button or other means) to start a covert enrollment procedure. In the covert procedure, a small audio fragment is recorded and the enrolled person is placed on a "to do" list so the user can assign a name to the enrolled person later. There should be options to divide (unnamed) persons into groups, but finding the exact requirements for this grouping functionality is a topic for future research.

8 Triggering and Reporting Identifications

This chapter discusses how FaceSpeaker should be triggered to identify a person, how the person's identity should be conveyed to the user and what the timing requirements for this procedure would be.

8.1 Triggering Identification

This section discusses the manner in which FaceSpeaker should be triggered to identify a person. The FaceSpeaker prototype identifies a person automatically as soon as he comes in view of the camera. However, most related prototypes and devices require the user to manually trigger an identification. Astler et al's device requires the user to point the camera (mounted on his long cane) at the person to be identified, and then requires the user to say "recognize" to trigger the identification [3]. Using speech recognition to trigger an identification is fine (and probably intended) for demonstration purposes, but is clearly too obtrusive and slow for use outside the laboratory. If I have to say "recognize" every time I need to identify a person I might as well do without the device and ask the person for his name.

Orcam's device requires the user to point at a person to trigger recognition [27]. This is more discrete, but it is probably still too obtrusive for use in a social interaction setting. In particular, a pointing gesture is normally used as nonverbal communication. Using a pointing gesture for other purposes can cause major social problems as illustrated by the below scenario.

Scenario 6 : pointing

This scenario serves to illustrate social acceptability issues which may arise from the need to issue a manual "recognize" command of some sort, such as the need to point at a person's face to trigger identification of that person.

This scenario assumes I'm using a device which identifies people if I point at them, and that I have enough residual vision to accurately point at people. I'm sitting at a table where some jokes are being swapped causing us to have a good laugh. I see a man approaching our table and I think I recognize him. So I look in his direction, raise my hand in front of my face to get it in view of the camera and point a finger at his face to get his identity. It turns out I didn't talk to the man, but he saw me make the gesture. He approaches our table and says : "Hi guys, I'm the organization committee chair. What is so funny about me?". A litle embarassing...

Ultimately, a fully manual approach requiring the user to take some action (beyond pointing the camera) every time a person should be identified seems unworkable. As pointed out in the PACT analysis, "identifying a person" is an extremely frequent activity in a social interaction context. Finding a sufficiently unobtrusive means to issue an "identify" command is challenging. Many methods create a risk of causing issues such as the problem illustrated in the previous scenario. In addition, control methods which are unobtrusive if used infrequently can become obtrusive if used too often. For example, occasionally pressing some button located on a wristwatch or in a trouser pocket will probably not be noticed. But if I press such a button every time I want to identify a person (which can be multiple times per minute) it might be perceived as a "tick" and annoy people around me.

Beyond the obtrusiveness problems, issuing a manual command places demands on the user's attention. As noted in the PACT analysis user attention is an extremely scarce resource in a social interaction context. Finally the need to issue a manual "identify" command inevitably increases the time needed to complete the activity of "identifying a person". Given the fact this activity should be completed in under a second (see the section on timing for more information about this), the extra delay caused by a manual "identify" command is both significant and highly undesirable.

While a "fully manual" identification procedure will not suffice in most situations, a nave "fully automatic" identification procedure such as implemented in the FaceSpeaker prototype is not acceptable either. The main problem is that constant automatic identifications distract the user too much. This was very apparent in FaceSpeaker field trials. Even in a demonstration setting, FaceSpeaker greatly disrupted the conversation by "cheerily" beeping and talking away as my conversation partner went into and out of the camera's view. What is required is a "filtered automatic" approach, in which FaceSpeaker automatically identifies persons without unnecessarily disrupting the user.

The FaceSpeaker prototype will repeat the same name over and over again as a face leaves and enters the camera's field due to factors such as head movements by the user. This is obviously unnecessary. Usually speaking the name once would suffice although users might sometimes need to have the name repeated (because they could not hear it properly over the background noise or they didn't pay attention). So the name should not be repeated if the same person reenters the camera field briefly after he left it, and a button or other simple method should be provided for the user to repeat the last speech message.

Another option to reduce the number of user interruptions is to filter identifications. The current prototype notifies the user when an unknown face comes in view of the camera. For many users those notifications are unnecessary distractions. Partially sighted users will usually see a person is standing in front of them and would not need FaceSpeaker to confirm this. Therefore, users who prefer not to be notified about unknown faces should be able to turn such notifications off so FaceSpeaker will only issue feedback if an enrolled person comes in view of the camera.

Beyond this simple configuration option an option to only be notified when certain persons are identified may be useful, particularly in situations where many enrolled people are gathered. A user might want to pick a single individual or an individual who belongs to a certain group out of the crowd. Properly assessing if such options are desirable in practice and detailing how they are to be implemented would require more elaborate field trials which are not yet possible given the limitations of FaceSpeaker and comparable devices. Therefore those proposals should be considered as general directions for future research.

8.2 Reporting Identifications

This section discusses the manner in which FaceSpeaker should report the identity of a person to the user. When a person comes in view of the camera, the current prototype first issues a beep. If the person is not enrolled yet, the beep is low pitched. If the person is enrolled, the beep is higher pitched (the higher the pitch, the more confident the identification is). The name of the identified person is spoken about half a second later. The beep mainly serves to shift the

user's attention towards this verbal utterance. When the person moves outside the camera's view, FaceSpeaker indicates this by another low pitched beep.

During FaceSpeaker field trials, a blind potential user suggested replacing the beep by a haptic (vibrotactile) signal in order to lower the demands on the user's auditory sense. As explained in the PACT analysis, the user's "auditory attention" is a scarce resource which may easily get overloaded in a social interaction setting. The idea of using haptic feedback was already explored in 2 related studies. Advanced Medical Electronics Corporation's proposed face recognition device for early stage Alzheimer patients [6] displays the name of the identified person on a wristwatch which vibrates to indicate the device has identified a person [6]. The ICare interaction assistant team has designed a vibrotactile belt for use with their device [20]. The main purpose of this belt is to convey the number, relative distance and position of other persons in a room to the user. In a follow-up publication they give a detailed description of experiments to determine the appropriate characteristics for the vibrotactile signals such as rhythm, frequency and duration [21]. As those topics are outside the scope of the present FaceSpeaker study they will not be discussed further here, but the reader is referred to the cited publications for more information.

Based on the cited papers the conclusion is that vibrotactile signals would be an effective means for FaceSpeaker to shift the user's attention to the verbal utterance of an identified person's name, and that it is possible to vary the vibrotactile signals such that they can effectively convey different situations such as an unknown face or various degrees of identification confidence. Investigating the details of such feedback mechanisms is a topic for future research.

8.3 Timing

The conclusion so far is that FaceSpeaker should automatically identify persons and apply some filtering to avoid unnecessarily disrupting the user. When a person comes in view of the camera, FaceSpeaker would first issue a vibrotactile signal to the user followed by a verbal utterance of the person's name. This section discusses the timing of those signals.

The main question when deciding on appropriate timing is the question of what constitutes a socially acceptable time for the activity of "identifying a person". Utsumi et al. investigated this issue [31]. They argue that the socially acceptable time limit for identifying a person is about 900 milliseconds. They reason that according to Thorpe et al, it takes an average of 445 milliseconds for a person to recognize and react to a complex visual scene [29]. So if I failed to recognize an acquaintance, he would begin responding to my lack of recognition after a minimum time of 890 milliseconds.

The magnitude of this time limit is debatable. In Thorpe et al's experiments subjects responded to the presence of an animal in an image which was briefly displayed to them. The task of spotting an animal cannot be compared to the task of identifying a human face. One reason is that the human brain contains systems specific to face recognition which are anatomically separate from systems for general object recognition [8].

This being said, Utsumi et als simple argument can be accepted as providing a useful model to give a rough estimate of the maximum time available for conveying the identity of a person to a FaceSpeaker user. If FaceSpeaker managed to convey a persons identity within 900 milliseconds , this would be

fast enough to be socially acceptable. Utsumi et als experiments suggest that a FaceSpeaker-like device could technically convey a persons identity to the user within 900 milliseconds using current algorithms.

However, they are assuming a device which would display the identified persons name visually. FaceSpeaker needs to convey this name through text to speech technology, which takes much more time than showing a text on a display. In addition some time might have to be factored in for shifting the users attention towards the verbal utterance of the name. FaceSpeaker trials suggest that if the name is spoken too abruptly, this may result in the user not understanding it due to a lack of attention. In addition, taking a bit more time may allow for more accurate face recognition algorithms or other mechanisms to improve Face Speakers reliability.

It is likely that (visually impaired) FaceSpeaker users cannot and will not be held to the same time limit for identifying a person as the general sighted population. Utsumi et als argument implicitly assumes that people issue nonverbal cues of recognition, and that people expect to be recognized first. Because visually impaired people are by definition severely restricted in their visual nonverbal communication abilities, their acquaintances cannot expect them to issue the same nonverbal cues of recognition which a normally sighted person would issue.

Probably a somewhat slower identification would be perfectly acceptable to acquaintances if this resulted in a higher chance of successful identification, but determining what would be an appropriate time limit would require more systematic experimentation. For a starting point, the below scenario draws an analogy to a situation where 2 normally sighted people who did not meet in a long time might not necessarily expect to be recognized first. This analogy suggests that for a FaceSpeaker user, a limit of 1.5 times the normal limit (or approximately 1400 milliseconds) would be reasonable.

Scenario 7: "Long time no see!".

This scenario serves to motivate the proposed "extended time limit" for identifying a person by which visually impaired persons would be judged. The rigid time definitions are not realistic and the thoughts made explicit in this scenario are normally subconcious.

Gerhard and Jonathan (both sighted) have met at another conference five years ago. They approach each other, and begin to look at each other at time t1 = 0 milliseconds.

Gerhard has a great memory for people. He instantly recognizes Jonathan and is positively surprised to see him again. At time t2 = 445 MS, his body language reflects the recognition and positive reaction. Jonathan did not yet recognize Gerhard, and has a blank expression on his face.

At time t3 = 890 MS, Gerhard still sees the blank expression on Jonathans face. He reasons: Jonathan still did not recognize me, so I would normally conclude he forgot me and put an offended expression on my face. But I realize we met 5 years ago and I am better at recognizing people than others. 445 milliseconds ago, I changed my body language to reflect recognition and positive surprise. This could serve as a cue for Jonathan, and he would have needed at least 445 milliseconds to spot this cue and react to it. Then I might need some time to spot his reaction. Well Ill give him another 445 milliseconds to show recognition.

If Gerhards display of recognition freshened up Jonathans memory, Jonathan

would show the same signs of recognition no later than time t4 = 1335 milliseconds. Gerhard find this perfectly acceptable under the circumstances indeed, he would be impressed that with a little embarrassing...years because he is aware of his unusually good memory for people.

For future prototypes, a suggested timing for the signals is to start the vibrotactile signal as soon as technically feasible so the user gets the maximum amount of time to shift attention to the verbal utterance of the person's name. Ideally, speaking the person's name should be timed such that the verbal utterance finishes no later than 1.4 seconds after the person came in view of the camera. However, keeping this 1.4 second limit could be challenging in case of a longer name. It is suggested that while starting the vibrotactile feedback as soon as possible and starting the verbal utterance of the name well before the 1.4 second mark are important requirements, it may not be a problem in practice if the verbal utterance does not finish before the 1.4 second mark. It seems more important that the FaceSpeaker user gets the name of an identified person right without the need for manual commands than to precisely stick to a certain time limit.

8.4 Conclusion

The requirements for triggering identifications and conveying a person's identity to the user are summarized as follows. When a person comes in view of the camera, FaceSpeaker should issue vibrotactile feedback to the user within the socially acceptable time limit for "identifying a person" of 900 milliseconds. This signal alerts the user that a face is in view of the camera, and can be varied to give the user additional cues For more information about vibrotactile feedback consult references [20] and [21].

In case of an enrolled person, FaceSpeaker will speak that person's name such that the user gets enough time to shift his attention towards this verbal utterance, while the verbal utterance preferably finishes by the proposed "extended time limit" for identifying a person of 1400 milliseconds. However, it is more important to maximize the chance of successfully conveying a person's identity to the user than to precisely stick to this time limit.

9 Conclusion

The developed FaceSpeaker prototype provides a useful proof of concept. It demonstrates that a FaceSpeaker device is technically feasible and can benefit users in practice. The open source openCV face recognition algorithms are not yet robust enough for use in a practicable device, but openCV is under active developments and the robustness of open source face recognition algorithms is likely to improve in the near future. The main problem is that openCV does not yet feature algorithms to compensate for the highly variable lighting conditions encountered when using FaceSpeaker while the face recognition algorithms cannot handle such lighting variations either. Commercial face recognition libraries are already available, and other studies suggest that those commercial algorithms provide a more robust alternative.

The main requirements for a FaceSpeaker device are unobtrusiveness and economizing on the user's attention. While it is desirable to ask people for permission before enrolling them into FaceSpeaker's database, this is not always feasible and a covert enrollment procedure should be available. The user should be able to enroll a person by pressing a covert button. A small audio fragment will be recorded which enables the user to assign a name to the enrolled person later. A function to divide enrolled persons into groups may also be desirable.

For triggering identification of a person, a "filtered automatic" approach is necessary. A "manual" approach as used in many related devices, where the user needs to trigger identification of a person by issuing some sort of "recognize" command, is not workable due to issues related to obtrusiveness and timing. When an enrolled person comes in view of the camera, FaceSpeaker should initiate vibrotactile feedback within the socially acceptable time limit for identifying a person of 900 milliseconds. After the user's attention has been engaged by this signal, FaceSpeaker should speak the identified person's name. This verbal utterance should preferably be finished within the proposed 1400 milliseconds "extended time limit" for identifying a person by which a visually impaired person would be judged, although it is more important to maximize the chance of accurate identifications successfully conveyed to the user than to strictly observe this time limit.

Follow-up research should focus on developing more robust prototypes which incorporate the design recommendations summarized above. This will enable more elaborate field trials to validate and elaborate on those design suggestions.

References

[1] AfterShokz. AfterShokz open ear bone conduction headphones. http://www.aftershokz.com.

[2] Joseph Albahari. Threading in c#. http://www.albahari.com/threading, 2006.

[3] Douglas Astler, Harrison Chau, Kailin Hsu, Alvin Hua, Andrew Kannan, Lydia Lei, Melissa Nathanson, Esmaeel Paryavi, Michelle Rosen, Hayato Unno, et al. Increased accessibility to nonverbal communication through facial and expression recognition technologies for blind/visually impaired subjects. In *The proceedings of the 13th international ACM SIGACCESS conference on Computers and accessibility*, pages 259–260. ACM, 2011.

[4] David Benyon, Phil Turner, and Susan Turner. *Designing interactive systems: People, activities, contexts, technologies*. Pearson Education, 2005.

[5] Charles Arthur. Google 'bans' facial recognition on Google Glass - but developers persist. *The Guardian*, june 3 2013.

[6] Advanced Medical Electronics Corporation. Face recognition device to aid early stage alzheimer patients, SBIR award93326. http://www.sbir.gov/sbirsearch/detail/76119, 2009.

[7] Emgu CV. Emgu cv: open cv in .net (c#, vb, c++ and more). http://www.emgu.com.

[8] Martha J Farah. Is face recognition special? evidence from neuropsychology. *Behavioural brain research*, 76(1):181–189, 1996.

[9] Hans Förstl and Alexander Kurz. Clinical features of alzheimers disease. *European archives of psychiatry and clinical neuroscience*, 249(6):288–290, 1999.

[10] Lakshmi Gade, Sreekar Krishna, and Sethuraman Panchanathan. Person localization using a wearable camera towards enhancing social interactions for individuals with visual impairment. In *Proceedings of the 1st ACM SIGMM international workshop on Media studies and implementations that help improving access to disabled users*, pages 53–62. ACM, 2009.

[11] Google inc. Getting to know Google Glass: Google Glass help pages. https://www.google.com/glass/help/#get-started.

[12] Google inc. Google Glass. http://www.google.com/glass.

[13] Luxand inc. Luxand, inc. website. http://www.luxand.com.

[14] c. Johnson. Emgu multiple face recognition using pca and parallel optimization. http://www.codeproject.com/Articles/261550/EMGU-Multiple-Face-Recognition-using-PCA-and-Paral#Main, 2013.

[15] Sreekar Krishna, Dirk Colbry, John Black, Vineeth Balasubramanian, Sethuraman Panchanathan, et al. A systematic requirements analysis and development of an assistive device to enhance the social interaction of people who are blind or visually impaired. In *Workshop on Computer Vision Applications for the Visually Impaired*, 2008.

[16] Sreekar Krishna, Greg Little, John Black, and Sethuraman Panchanathan. A wearable face recognition system for individuals with visual impairments. In *Proceedings of the 7th international ACM SIGACCESS conference on Computers and accessibility*, pages 106–113. ACM, 2005.

[17] Sreekar Krishna and Sethuraman Panchanathan. Assistive technologies as effective mediators in interpersonal social interactions for persons with visual disability. In *Computers Helping People with Special Needs*, pages 316–323. Springer, 2010.

[18] Xiaoguang Lu, Yunhong Wang, and Anil K Jain. Combining classifiers for face recognition. In *Multimedia and Expo, 2003. ICME'03. Proceedings. 2003 International Conference on*, volume 3, pages III–13. IEEE, 2003.

[19] John Markoff. Device from israeli start-up gives the visually impaired a way to read. *the New York Times*, june 4 2013.

[20] Troy McDaniel, Sreekar Krishna, Vineeth Balasubramanian, Dirk Colbry, and Sethuraman Panchanathan. Using a haptic belt to convey non-verbal communication cues during social interactions to individuals who are blind. In *Haptic Audio visual Environments and Games, 2008. HAVE 2008. IEEE International Workshop on*, pages 13–18. IEEE, 2008.

[21] Troy L McDaniel, Sreekar Krishna, Dirk Colbry, and Sethuraman Panchanathan. Using tactile rhythm to convey interpersonal distances to individuals who are blind. In *CHI'09 Extended Abstracts on Human Factors in Computing Systems*, pages 4669–4674. ACM, 2009.

[22] Peter Meijer. FlycamOne2 head-mounted webcam for the bling. `http://www.seeingwithsound.com/flycamone2.htm`.

[23] Peter Meijer. the vOICe: seeing with sound, Augmented Reality for the Totally Blind. `http://www.seeingwithsound.com`.

[24] Peter Meijer. USB voltage booster: power supply for 9V camera glasses. `http://www.seeingwithsound.com/usb_powersupply.htm`.

[25] Erez Na'aman, Amnon Shashua, and Yonatan Wexler. Patent ep2490155a1. a user wearable visual assistance system, august 22 2012.

[26] openCV. opencv website. `http://www.opencv.org`.

[27] Orcam. Orcam website. `http://www.orcam.com/`.

[28] SBIR. About the SBIR (Small Business Innovation Research) program. `http://www.sbir.gov/about/about-sbir`.

[29] Simon Thorpe, Denis Fize, Catherine Marlot, et al. Speed of processing in the human visual system. *nature*, 381(6582):520–522, 1996.

[30] ICCHP-Summer University. ICCHP summer university on mathematics, science and statistics website. `http://www.icchp-su.net`.

[31] Yuzuko Utsumi, Yuya Kato, Kai Kunze, Masakazu Iwamura, and Koichi Kise. Who are you?: A wearable face recognition system to support human memory. In *Proceedings of the 4th Augmented Human International Conference*, pages 150–153. ACM, 2013.

A Camera

Various options for a body mounted camera have been investigated. This chapter first discusses the best position for a body mounted camera and then describes the various alternatives which have been investigated.

A.1 Camera Position

The camera must be positioned such that it has a good chance of capturing faces and is unobtrusive. Most related prototypes have used camera glasses, and so will FaceSpeaker. As will be shown in the rest of this chapter, various high quality camera glasses are commercially available as of 2013. A camera mounted at eye level is in a good position to unobtrusively capture other people's faces. When 2 persons interact, they commonly look at each other's face. This means that moving one's head such that the eyes (and hence the camera embedded into the glasses) is pointed at another person's face is a natural, socially accepted action. Another reason for using camera glasses is the emergence of "augmented reality" glasses such as Google Glass (see section 2.3 for details). As such products penetrate the market, wearing camera glasses will become more socially accepted.

While most cited devices propose using camera glasses (see section 2.1), Astler et al's successful face recognition device is a notable exception [3]. Initially, they planned to use camera glasses. However, apparently they did not find glasses featuring a camera which suited their quality requirements. They do not specify exactly which options they have considered, but it should be noted that miniature camera technology has developed at a dramatic pace over the last years so the quality of the camera glasses available at the time of their research probably did not approach the quality of camera glasses available as of 2013. In addition, all camera glasses they considered were apparently sunglasses. Their user surveys demonstrated that many users were not comfortable wearing sunglasses. Sunglasses may prevent partially sighted users from utilizing their residual vision in many (indoor) situations. In addition, wearing sunglasses indoors may decrease the social acceptability of a face recognition device. The FaceSpeaker author agrees that sunglasses are not acceptable in most usage scenario's including the small conference scenario, especially when the user is partially sighted. However, the Lawmate camera glasses used by FaceSpeaker can be fitted with transparent or prescription lenses eliminating this problem. Finally Astler et al. considered mounting a small camera atop glasses, but this option was discarded as being too obtrusive. FaceSpeaker's findings confirm this conclusion; as will be demonstrated later in this chapter, field trials using a clearly visible head mounted camera captured a great deal of attention and elicited strong social reactions.

After considering various other options including a chest mount, Astler et Al. decided to mount a camera atop the blind user's long cane. This enabled them to use a larger camera of higher quality. This solution was generally effective. Their device (especially the face recognition part) worked well and users were able to point the camera at other persons. Most users were quite positive about the cane mount, though some users raised objections and would have prefered another solution. In summary, a cane mounted camera is a good option which may have certain advantages over camera glasses. However, a cane mounted

camera is arguably not a good option for most users in the context of the small conference scenario. Long canes are only used by the totally blind or people with extremely restricted vision. Most partially sighted users will use a short cane (for recognition in traffic). In the author's experience, short canes are rarely if ever used indoors and even long canes are typically put away during indoor social interactions. Now that high quality transparent camera glasses are available this seems a better option, although it is acknowledged that a cane mounted camera will have quality advantages and might in some cases be easier to point at people's faces. In any case, the FaceSpeaker software should work using the camera setup Astler et al. describe if the user prefers that option.

A.2 Lawmate Camera Glasses

The Lawmate camera glasses used by FaceSpeaker are 5 megapixel analogue camera glasses which can be fitted with sunglass, transparent or prescription lenses. The camera is optimized for indoor and outdoor use under variable lighting conditions. The design is professional. The camera is hidden in the frame's right arm and is practically invisible even at short range. The camera glasses are typically used by journalists or other people who need to unobtrusively make television-quality recordings.

A.3 Alternative Camera Glasses

Many camera glasses are available on the market. Typically, those glasses are intended for recording sporting or other outdoor activities. The recorded video is stored onto internal memory or a micro SD memory card. Many such glasses feature a USB port for transferring the recorded video to the computer, but few such glasses feature a USB webcam function. However there are a few exceptions.

Figure 11: USB video sunglasses used by "the vOICe". Source: seeingwithsound.com [22]

"the vOICe" provides various suggestions for affordable camera sunglasses which can be connected directly to a laptop using a USB cable and will function like a normal webcam [23]. Compared to the Lawmate glasses, the camera and cable are much more obtrusive and it is not possible to fit those glasses with transparent lenses. In addition, they are "tourist style" sunglasses which might not be appropriate to wear in the context of the small conference scenario. Nevertheless they do provide a good option for easy experimentation with FaceSpeaker and other prototypes.

A.4 Camera on Headband

A camera mounted on a headband has been considered. This is a very obtrusive option, and it is not realistic outside an experimental setting. However, had a headband mounted camera proven to be a good option there might have been possibilities to hide a camera in a hat or other fashion item. But after the discovery of the Lawmate glasses and unsatisfactory field trials using such a camera, this option was abandoned.

A.4.1 Flycam One2

"the vOICe" suggests the Flycam One2 as a headband mounted camera for use with their software [22]. This camera is typically mounted to a model aircraft. Besides the option to record to a secure digital memory card, this camera features a function to use it as a standard USB webcam. One of the accessories is a headband. A FlycamOne2 was ordered and tested.

Unfortunately, it turned out no 64 bit webcam driver was available preventing the author from testing it. The author made some test recordings using the built-in recording function and was disappointed by the quality. Therefore the Flycam one2 was abandoned.

Figure 12: FlycamOne2 mounted on a headband. Source: seeingwithsound.com [22]

A.4.2 Standard Logitech Webcam

A Logitech Pro9000 webcam was disassembled and mounted onto the Flycam one2's headband using 2 tie wraps (one of them running through the holes for the hinging mechanism at the back of the camera, the other running over the camera). The below image demonstrates this setup.

The author performed a field test using an early FaceSpeaker software version and this camera setup. As expected the head mounted camera elicited strong social reactions. When people (even strangers) saw this setup they would often immediately start asking questions about it. When the FaceSpeaker project was explained they were generally positive and curious, however they were clearly uneasy about a camera mounted in this manner. Soon, it was obvious the pro 9000 could not handle the variable lighting conditions and movements inherent in head mounted use. This webcam is designed to be mounted on a desktop computer monitor. The camera can then record the user's face and other objects provided lighting conditions are relatively stable. When the camera was head mounted, very few faces could be detected let alone trained or identified. Outdoors the camera was not functioning at all.

Yet another problem was aligning the camera to the eyes. In order to get any results without having to make strange head movements the headband had to be stretched far to get the camera right above the eyes. The page describing the use of the FlycamOne2 with the vOICe notes the same problem and suggests turning the camera upside down [22]. For face recognition, aligning the camera with the user's eyes may be even more important than for the vOICe. FaceSpeaker needs to capture faces (which are typically near the user at eye level) with as little rotation as possible. This need for alignment with the eyes significantly limits the options for mounting a camera into a headband or into another fashion item.

Figure 13: The FaceSpeaker author wearing a Logitech Pro9000 webcam mounted onto a headband

Overall this headband experiment yielded disappointing results. With the discovery of the Lawmate camera glasses, the option of using a head mounted webcam was abandoned. It is possible that a newer, higher quality webcam would have yielded better results, but the problems related to camera allignment suggest that even in this case a head mounted camera is not a good option.

B Frame Processing, Multithreading and Performance Optimization

This chapter describes how FaceSpeaker processes frames, takes advantage of multicore processors and manages power consumption. This is a technical chapter which assumes the reader has some programming background.

Detecting and recognizing faces in a live video stream is a computationally intensive task. Therefore, FaceSpeaker needs to work efficiently and must take advantage of all the processing power the computer has to offer. Nearly all modern computers feature a multi core central processing unit (CPU). Taking advantage of those multicore CPU's requires a multithreaded application.

Programming a multithreaded application can be a challenging task and requires specialized knowledge about distributed programming. Fortunately, the c# programming language and the Microsoft .net framework 4.0 contain many features to make writing multithreaded applications easier [2]. Those features were a key reason for choosing the c# programming language.

Nevertheless, getting FaceSpeaker to work reliably in a multithreaded environment proved complicated and took up much of the development time. This chapter explains some of the design choices and resolved issues related to multithreading and performance optimization.

B.1 Frame Capturing and Processing

FaceSpeaker's "operation loop" can be summarized as follows. Images captured by the camera are wrapped in a "frame object" and the object is stores in a "frames to process" queue. Every time a frame is being added to the FramesToProcess queue, a new task "processFrame" is started using c#'s thread pooling mechanism. This mechanism automatically and efficiently divides all tasks over multiple threads [2]. Thanks to this mechanism multithreaded computations run efficiently while the programmer doesn't need to worry about most of the technical complications involved.

Every "processFrame" task dequeues a frame from the FramesToProcess queue, detects the faces present on that frame and recognizes those faces. It then puts the processed frame object (now containing identifications for the faces present) in the RecentFrames sorted list.

In addition, a "onNewFrame" event is fired whenever a new frame has been captured. The user interface will display the new frame's image and attempt to identify the person currently in view of the camera (using the identifyPerson function) approximately every second.

B.2 Frame Rate Throttling

Most cameras can capture 25 to 30 frames per second, (fps), but most personal computers in use as of 2013 cannot perform live face recognition using the OpenCV algorithms at that frame rate even if the program is multithreaded.

Therefore FaceSpeaker automatically throttles the frame rate if frames are being processed too slowly. Frames should be processed within half a second after they have been captured.

If a ProcessFrame task detects that processing a frame finishes over half a second after that frame has been captured, it processes the frame normally

but lowers the camera's frame rate by 1 fps. There is at most one frame rate decrease per second to prevent excessive throttling.

If a frame has been waiting for processing in the FramesToProcess queue for over half a second it is not processed anymore. This avoids an accumulation of unprocessed frames in the FramesToProcess queue which would degrade performance even further. If within the last second no frames have exceeded the processing deadline, the camera's frame rate is increased by 1 fps again. This prevent temporary performance inhibitors from causing excessive frame rate throttling.

This mechanism avoids wasting system resources on capturing frames which cannot be processed in time, while avoiding excessive frame rate throttling.

B.3 Power Saver

The high CPU utilization caused by running FaceSpeaker is especially problematic when it is used on a laptop computer. High CPU utilization causes increased power consumption and heat production. This will drain the laptop battery very quickly, and the excessive heat - which cannot escape the user's backpack - may cause significant discomfort to the user or damage to the laptop.

When a person is in view of the camera, a high frame rate is desirable because this increases the probability of detecting a usable face and increases recognition accuracy. However, when no face is in view of the camera or when a person has been recognized and stays in view of the camera for an extended period of time, the high frame rate just wastes CPU resources.

Therefore a power saver feature was implemented. If enabled, FaceSpeaker enters standby mode if it did not detect a face for over 2 seconds or if a person has been finally identified (e.g. the user interface has issued the second beep indicating recognition confidence). In standby mode the capture delay is fixed at 200 milliseconds, causing the frame rate to drop to 4 frames per second. This is enough to detect when a person comes into view (and as soon as a face is detected FaceSpeaker will leave standby mode). The reduced frame rate causes a considerable drop in CPU utilization without having a noticeable impact on FaceSpeaker's performance.

B.4 Process and Thread Priority

In early field trials 2 symptoms surfaced:

- Sound playback and speech output were delayed or distorted.
- When a screen reader was active, the screen reader tended to "lock up". This made it nearly impossible to read or navigate the user interface.

This is a case of resource starvation. In a multitasking operating system such as Microsoft Windows, the machine runs multiple processes (programs) simultaneously. Each process consists of one or more threads. A part of the operating system called the "scheduler" is responsible for distributing available CPU time among the various processes and threads trying to perform calculations. When multiple processes are active the scheduler should distribute the available CPU time equally among processes and should distribute each processes' CPU time equally among the processes' threads. However in practice the scheduler's operation is not that simple.

Notably, the scheduler operates in a non-deterministic fashion. This means that a programmer cannot make any assumptions as to how the scheduler will distribute available CPU time among competing processes and threads. In other words, the programmer has no way of estimating what amount of CPU time his program gets in a given time frame, or how CPU time will be distributed among the processes' threads. If CPU time is scarce, process / thread execution will inevitably be delayed in order to allow other threads to execute. Those waiting delays will be of unpredictable, highly variable duration. FaceSpeaker's background threads (e.g. the threads responsible for face detection and face recognition) request an inordinate amount of CPU time, indeed more CPU time than most computers can provide. Behind the scenes, c#'s thread pooling mechanism distributes the "background" work over many background threads. The "user interface thread" (the thread responsible for interacting with the user and issuing messages) is just one of n (n ¿¿ 1) threads competing for CPU time on the fully occupied CPU.

Due to the non-deterministic operation of the scheduler there will be unpredictable and highly variable delays in the execution of the user interface thread. If the CPU is fully occupied, those execution delays can be long enough to cause noticeable effects in the user interface such as delayed or distorted sound and speech. Because the FaceSpeaker process tries to claim more CPU time than is available, competing processes (such as screen readers) may experience similar noticeable delays.

Fortunately, process / thread priority allow programmers to have the scheduler prioritize certain processes / threads over (or below) others when distributing CPU time. FaceSpeaker's user interface thread has a thread priority of "high", meaning the scheduler will assign this thread more CPU time than all other threads (which have the default thread priority of "normal"). The FaceSpeaker process as a whole has a process priority of "below normal". This prevents it from starving other processes such as screen readers of CPU time. After setting those priorities, the symptoms described earlier both disappeared while FaceSpeaker retained about the same frame rate.

B.5 Improving Performance

After implementing the mechanisms described in this chapter and resolving many subtle technical issues, FaceSpeaker worked well enough for a prototype product. However, given enough development time and resources FaceSpeaker's performance could be improved.

As described on EmguCV's licensing page, a commercial version of EmguCV is available. This has better performance characteristics and native multithreading support thanks to a more optimized build process. At about $200 this product is quite affordable Although it would still have had a significant impact on FaceSpeaker's very limited budget. Because FaceSpeaker performs acceptably using EmguCV's open source release and using a commercial EmguCV version might interfere with FaceSpeaker's goal of releasing an open source product, this option was not investigated further. But if FaceSpeaker were to be developed beyond the current experimental phase, designing an optimized build process might be an easy way of improving performance.

Changing the process and thread priorities as explained in the previous section effectively resolved the problems experienced, but this quick fix is not an

ideal solution. Lowering FaceSpeaker's process priority will cause it to yield significantly more CPU time to background processes. The work done by those background processes will always slow down FaceSpeaker but does not necessarily benefit the user directly (most users would prefer better FaceSpeaker performance over background processes like virus scans). Albahari points out that prioritizing the user interface thread over other threads may cause user interface updating code to take up a disproportionate amount of CPU time, unnecessarily slowing down the background threads [2]. The solution Albahari proposes is to split up the application into 2 processes. One process is responsible for running the user interface thread, the other process is responsible for running all other threads. The 2 program parts would then communicate through inter-process messaging.

This is a desirable solution for various reasons. Not only could it allow significant performance increases without fiddling with priorities, but once this solution has been implemented it becomes relatively easy to offload the face detection and recognition algorithms onto a server (which would basically receive the low bandwidth grayscale camera stream through a mobile internet connection and pass back the negligibly small recognition results to the client). The best part of such a solution is the possibility to run FaceSpeaker on augmented reality glasses like Google Glass or other mobile devices available today. Again such options are outside the scope of prototype development but if FaceSpeaker is to be developed further this should be a top priority. The user interface is loosely coupled to the rest of the program, so it is relatively straightforward to write some classes bridging the user interface and program backbone through a network connection. Of course a user interface suitable for augmented reality glasses or other mobile devices must also be developed.

C FaceSpeaker Software User Guide

This manual describes the FaceSpeaker software and explains how to use it. The software can be downloaded from the FaceSpeaker website.

Access Keys

Every focusable component of the FaceSpeaker user interface has an access key (alt +a letter or symbol). This allows the user to quickly navigate the user interface without using a mouse. If the access key is pressed, focus moves to the element. In case the target element is a button the button is immediately activated and focus does not move; pressing a quick access key for a checkbox moves focus to the checkbox and toggles its state. Throughout this documentation, access keys will be listed in brackets next to each element (e.g. "trained persons list [alt + p]"). A table listing all focusable elements and their access keys can be found at the end of this manual.

Important Notes

FaceSpeaker can only recognize people optimally if ¡strong¿at least 2¡/strong¿ persons are in the database; by default FaceSpeaker comes trained with the author's face. On some laptops, power management software may throttle CPU speed when the machine is running on batteries to conserve power. FaceSpeaker is computationally intensive so it may not work optimally if the CPU speed is throttled. Therefore, please ensure your power scheme does not inhibit CPU performance (the easiest way to do that is to change your power scheme to "maximum performance"). FaceSpeaker does not utilize the GPU (graphics chip).

Starting FaceSpeaker

After downloading and installing FaceSpeaker, start it using the shortcut placed in the start menu's programs or apps section. The FaceSpeaker window will appear, and FaceSpeaker will say "loading data...". Once the database of trained persons has been successfully loaded, FaceSpeaker will say "initializing camera and face engine...". Once everything has been loaded and initialized, FaceSpeaker will issue a sound and say "starting capture!". The time FaceSpeaker needs to initialize depends on the number of persons in the trained persons database and the computer's performance characteristics. During the initialization process FaceSpeaker will issue a sound every 3 seconds to let the user know it is still initializing. Once FaceSpeaker is loaded, the grayscale video feed captured by the camera will be displayed in the top right part of the FaceSpeaker window.

Adding a Person

To add a person to the trained persons database, enter the name of that person in the "name" textbox [alt + n] and click the "add person" button [alt + a]. FaceSpeaker will say "start training!".

Make sure the person to be added is in front of the camera, and make sure no 2 faces are in the camera's field of vision. If FaceSpeaker detects a face in front of the camera, it will say "face found!" and begin capturing faces after a short delay. Every time a face has been added, FaceSpeaker issues a click sound. If FaceSpeaker does not detect a face in front of the camera, it will issue a sound and say "waiting..." at 3 second intervals until a face is detected in front of the camera.

After 20 faces have been added, FaceSpeaker will say "Done! Retraining face engine." When the face engine has been retrained, FaceSpeaker will issue a sound and say "starting capture!".

Recognizing Persons

After FaceSpeaker has said "starting capture" and the person just added is in front of the camera, FaceSpeaker should recognize that person. When a person is recognized, FaceSpeaker first issues a short beep. It then speaks the name of the person detected. Assuming the detected person stays in front of the camera, FaceSpeaker issues a short beep after approximately one second. The pitch of that beep indicates the recognition confidence: the higher pitched the beep, the more likely it is that the person has been correctly recognized. If the second beep is not issued, it indicates the recognized person is no longer detected in front of the camera; this often suggests the recognition is not reliable. If FaceSpeaker detects an unknown face in front of the camera, it issues a long low pitched beep. sound. If FaceSpeaker detects there is no longer a face in front of the camera, it issues a long low beep to indicate this to the user.

Managing Trained Persons

The "Trained persons" listbox [alt + p] contains the names of all persons in the trained persons database. You can select a person and perform various actions for that person.

Adding Training Data

Usually, one training session is not sufficient for FaceSpeaker to reliably recognize a person under lighting conditions different from the lighting conditions at training time. If a previously trained person is misidentified or recognized as unknown, training data must be added for that person. To add training data, select the person in the trained persons listbox [alt + p] and click the "add training data" button [alt + t]. FaceSpeaker will capture and add faces in the same manner as described in the "adding a person" section, except that in a retraining session only those faces which are incorrectly recognized based on existing training data will be added. This way, the retraining procedure ensures the trained persons database will contain a broad variety of face images acquired in different situations while the database does not get filled up with a lot of duplicate images. The retraining procedure continues until 20 faces have been added or 5 faces have been captured and correctly recognized using existing training data (which suggests further retraining under current conditions is not useful).

Deleting a Person

A person can be deleted by selecting his name in the "Trained persons" listbox [alt + p] and pressing the "delete person..." button [alt + d]. FaceSpeaker will display a dialogue asking the user to confirm or cancel the deletion. Press [enter] to permanently delete the selected person or press [Escape] to cancel this action.

View Person Details and Trained Face Images

The bottom left part of the FaceSpeaker window displays information about the user selected in the "Trained persons" listbox [alt + p]. The read-only "Selected person details" multiline text box [alt + s] lists the person's name and the number of trained faces saved to the trained persons database. Below this text box is the "trained face images" group box. This group box allows the user to visually review the faces stored for a person. Press the "¡" button [alt + ¡ meaning alt + shift + comma] to view the previous image or the "¿" button [alt + ¿ meaning alt + shift + dot] to move to the next image. In future FaceSpeaker versions this functionality will be expanded and the user will be able to delete bad face images manually.

Power Saver

The power saver function limits the frame rate to about 4 frames per second if no face is detected for 2 seconds or if a person stays in front of the camera after having been finally identified. This lowers CPU utilization in order to decrease power consumption and heat production. It should not influence recognition accuracy but may make it more difficult to get the camera pointed at a face. The power saver function is on by default. It can be turned off by unchecking the "Power saver" checkbox [alt + o].

Frame Rate

The "Frames per second" read only textbox [alt + f] displays a constantly updated approximation of the camera's frame rate. FaceSpeaker automatically adapts the frame rate to the computer's performance, so if power saving mode is switched off the frame rate gives a rough indication of the computer's performance. The maximum possible frame rate depends on the camera and is usually 25 or 30 frames per second.

Getting Help

The "FaceSpeaker online help..." button [alt + h] opens this documentation. The "program version"read only textbox [alt + v] displays the program version. If you ask for support make sure to include the program version in your message.

Recognition Accuracy

For best results, use FaceSpeaker in good lighting conditions. There should be enough light and it should be uniformly distributed. Also ensure good camera quality and positioning. Good recognition requires "frontal face images". This means persons should look straight at the camera and the camera should not

be tilted relative to the face. People wearing glasses can often be normally recognized, but sunglasses interfere with face recognition.

The video feet acquired by the camera is being displayed in the top right part of the FaceSpeaker window, so a sighted individual can easily check if the images captured by the camera have good quality characteristics. The requirements described in the previous paragraph are not absolute, FaceSpeaker is designed to work under suboptimal conditions encountered in the "real world" when using a body mounted camera. However, keeping the guidelines in mind will help the user get better results and facilitate a fair judgment of FaceSpeaker's performance. Note that training FaceSpeaker to recognize a person in extremely bad conditions (a person wearing sunglasses, an extremely dark environment where the face is hardly discernible on the camera feet etcetera) will often be possible, but can influence recognition results under better conditions. Notably, FaceSpeaker may recognize other (unknown) persons as the person for whom training data was added under unfavorable conditions. Deleting and retraining that person under better conditions should resolve the issue.

Components and their Access Keys

The below table lists all focusable components of the FaceSpeaker user interface along with their access keys.

Component	**Access key**
Name text box	alt + n
Add person button	alt + a
Trained persons listbox	alt + p
Add training data button	alt + a
Delete person... button	alt + d
Previous image button	alt + ¡ meaning alt + shift + comma
Next image button	alt + ¿ meaning alt + shift + dot
Speech message textbox	alt + m
Approximate frames per second textbox	alt + f
Power saver checkbox	alt + o
FaceSpeaker online help... button	alt + h