Pan Shot Face Unlock: Towards Unlocking Personal Mobile Devices using Stereo Vision and Biometric Face Information from multiple Perspectives

Rainhard Dieter Findling

MASTERARBEIT

eingereicht am
Fachhochschul-Masterstudiengang

Mobile Computing

in Hagenberg

im September 2013

© Copyright 2013 Rainhard Dieter Findling

This work is published under the conditions of the *Creative Commons License Attribution–NonCommercial–NoDerivatives* (CC BY-NC-ND)—see http://creativecommons.org/licenses/by-nc-nd/3.0/.

Declaration

I hereby declare and confirm that this thesis is entirely the result of my own original work. Where other sources of information have been used, they have been indicated as such and properly acknowledged. I further declare that this or similar work has not been submitted for credit elsewhere.

Hagenberg, September 2, 2013

Rainhard Dieter Findling

Contents

Acknowledgements

The author wants to thank Dr. René Mayrhofer for his extensive support prior and during the research conducted for this thesis at u'smile, the Josef-Ressel Center for User-friendly Secure Mobile Environments, and for his steady support at introducing the author to the scientific community, without which this research would not have been possible. The author also wants to cordially thank his mother and father, Helga and Johann Findling, and his brother, Ronald Findling, for the extensive support, cantilevered discussions and useful input over the past years, especially during hard times.

Further, the author wants to thank his colleagues at u'smile (in alphabetical order): Michael Hölzl, Muhammad Muaaz and Peter Riedl for all their useful input and discussions with the author, Fabian Wenny and Dr. Clemens Holzmann for their support and cooperation during the research on range face detection and segmentation, Christopher Gabler for helping with the creation of the 2013' stereo vision pan shot data recording for the u'smile face database and Christina Aigner for helping at the creation of the initial, preliminary Hagenberg pan shot face database. Finally, the author wants to thank Dr. Christoph Schaffer for all the opportunities provided during the studies of Mobile Computing and his frequent concessions in order to make this research and thesis possible as they are.

This work has partially been carried out within the scope of *u'smile*, the Josef Ressel Center for User-Friendly Secure Mobile Environments. We gratefully acknowledge funding and support by the Christian Doppler Gesellschaft, A1 Telekom Austria AG, Drei-Banken-EDV GmbH, LG Nexera Business Solutions AG, and NXP Semiconductors Austria GmbH. Parts of this thesis have previously been published or are submitted for review in [62–65].

Abstract

Personal mobile devices hold a vast amount of private and sensitive data and can e. g. be used to access services with associated cost. For security reasons, most mobile platforms therefore implement automatic device locking after a period of inactivity. Unlocking them using approaches like PIN, password or an unlock pattern is both problematic in terms of usability and potentially insecure, as it is prone to the shoulder surfing attack: an attacker watching the display during user authentication. Hence, face unlock – using biometric face information for authentication – was developed as a more secure as well as more usable personal device unlock. Unfortunately, when using frontal face information only, authentication can still be circumvented by a photo attack: presenting a photo/video of the authorized person to the camera. In this work we present a variant of face unlock which is harder to circumvent than with using frontal face information only by using more facial information, available during a 180° pan shot around the user's head. We develop and evaluate our mobile device pan shot face unlock in four different stages in order to identify conceptual weaknesses and do improvements within the next stage. In the first stage we present a proof-of-concept prototype based on Android, which uses different Viola and Jones Haar-cascades for face detection and Eigenfaces for face recognition. We identify Eigenfaces as being insufficient for usage in a mobile device unlocking scenario. Therefore, we utilize neural networks and support vector machines for face recognition in the next stage, with which we identify using Viola and Jones based face detection as being insufficient for usage in a mobile device pan shot unlocking scenario based on multiple perspectives. Hence, we develop a novel face detection and segmentation approach based on stereo vision and range template matching in the next stage, which we find to deliver promising results and consequently focus on improving details of the range template generation and matching within the fourth and last stage. Parallel to developing and evaluating our approach we build up the u'smile face database containing grayscale and stereo vision pan shot test data. Concluding, our results indicate that a mobile device pan shot face unlock is a viable approach to unlocking mobile devices and that using range information might in general be an effective approach for incorporated face detection and segmentation.

Chapter 1

Introduction

1.1 Why Privacy and Authentication Matter on Personal Mobile Devices

Nowadays many people carry a mobile device – such as a smart phone – which has access to a large amount of data. In general, a notable amount of this data is considered to be private and deserves protection, such as a) information stored in messages such as mail, SMS, MMS or from instant messaging services, documents, pictures, videos and music stored on the device and cached data such as browser history, b) context related data, such as the current position (e.g. from GPS receiver or assisted, as with Wifi or mobile cell ID fingerprinting) and data from sensors included in the mobile device, such as acceleration sensors or gyroscope, c) information related to accessing a service or network, such as login data to private or company networks using e.g. VPN or Wifi, login data to mail services, websites and portals and even payment related information, such as access to banking, transactions and electronic forms of money (e. g. Falaki et. al. [60], Fried [74], Furnell et. al. [76]).

In case of this data falling into the hands of an unknown observer, a number of threats are possible: the observer could gain insight to private and classified information or could derive such information. They could further make use of it, e. g. of information related to payment services to conduct malicious transaction, or they could sell it to third parties. Moreover, the observer could assess behavioral patterns and predict future behavior, e. g. by performing location tracking and predicting future locations. Additionally, the observer could use the access to services to spread information in the device owner's name, or in order to perform account hijacking (taking over an account so that the legitimate user has no further access to it). Finally, the observer could use access to private and company networks to gain access to further data and devices.

In order to protect access to this data stored on a personal mobile device, access to the device itself has to be protected. In general there are two ways of accessing a mobile device: remotely and locally. Remote access means access without physical contact to the device and can be gained over a network e. g. via software accessing the network legitimately, or using an exploit for software installed on the device. Local access means access with physical contact to the device, such as the user interacting with the device directly. On the one hand, remote access can be limited or even refused, as it might not be necessary for the legitimate user locally interacting with the device. On the other hand, local access is necessary for the legitimate user to interact with this device. For this reason, and for mobile devices being lost or stolen much easier than classical desktop computers, protecting local access to the device is a very important task. As an example, even a short time of physical access to a personal mobile device might enable an attacker to install malicious software – which could grant the attacker remote access in the future, without the legitimate user even noticing the device as lost, stolen or contaminated. Therefore, this work is targeting the protection of local access to a personal mobile device against unauthorized users.

1.2 Security's Usability

End user security measures in combination with frequent device usage suffer a major drawback: they don't get applied voluntarily if their usability is too low. The problem especially with frequent device usage and local access protection is simple: from a user point of view, the positive effects of security are outperformed by the negative ones. For example, users facing less risk of somebody else accessing their private data will still not apply the therefore necessary security in case they are required to remember a long and complex password – and a few extra seconds during login when entering this password, each time they want to interact with the device. There are a few well known examples: studies show that if users are required to apply a password, but are free at choosing it, they most frequently choose short or rather incomplex and easy memorizable passwords [20, 90, 147, 200, 207]. This enables possible attackers to eventually derive the password from previously aggregated information about the user, or simply brute force it. In case of users applying a complex password, there commonly is the phenomenon of "cognitive load": as users already have to remember a single, long and complex password, they are likely going to apply this password wherever possible. Consequently, attackers are able to access de facto all of the user's services and devices once this password has been leaked for an arbitrary reason. These effects can be observed e. g. with company passwords, which widely only get changed frequently and with the required strength (in terms of length and complexity), if a corresponding policy is applied.

All these mechanisms apply to the mobile domain as well – with the extension that users don't actively use their devices contentiously, but stop and continue the interaction frequently. When applying security, this practically leads to a frequent locking and unlocking of the device – where of course security's bad usability preponderates. Consequently, security is not applied very widely on mobile devices, as stated e. g. in [14, 45, 48, 136, 201]. For this reason it is important that new ways of authenticating legitimate users with their personal mobile devices are developed, which provide convenient security not making the user feel uncomfortable at the same time.

1.3 Objective and Thesis Structure

We propose a pan shot face unlock for mobile devices, which a) uses more information than frontal face information only based on a pan shot of the device around the user's head and b) intends to be more secure and usable than current mobile device unlocking approaches. Our approach requires a mobile device with a front side camera and integrated gyroscope sensor, as it conceptually uses data recorded by cameras and sensors during a pan shot. In terms of cameras, a pan shot face unlock can make use of mono and/or stereo cameras. Using stereo cameras has the advantage of having range visual data available along with colored visual data. The aim of our approach is to a) increase security over current mobile device authentication approaches and b) still retain a high usability by a fast authentication and device unlock. Compared to other face authentication approaches, our approach requires more information than available in a photo or video of a face from a single perspective. Attackers would be required to construct a 3D model or obtain a closely synchronized video stream of the legitimate user in order to successfully conduct a photo attack (see section 2.3.4). As such data is harder to obtain than images showing a user's face from single perspective (which often can be obtained from social networks), our approach is conceptually harder to attack by a photo attack.

We review the currently most widely used classical and most important biometric authentication approaches for mobile devices and their conceptual problems in chapter 2. In chapter 3, we provide an overview of related work with a) research on mobile authentication systems, focused on face authentication, b) approaches to face detection and face segmentation and c) approaches to face recognition. We explain building blocks required for our method in chapter 4. In chapter 5 we describe our approach in detail with data aggregation, stereo to range conversion, performing face detection and face segmentation, performing face recognition and combining classifiers recognition results. We present the u'smile face database and its predecessor as source of test data to our approach in chapter 6. In chapter 7 we present the implementation and evaluation results of our approach in four different stages. Finally, we conclude and provide an outlook in chapter 8.

Chapter 2

User Authentication on Mobile Devices

Three basic factors are involved in user authentication (ensuring/confirming the identity of a person that wants to act as user of a certain system): knowledge, possession and inherence. When using knowledge based authentication, users authenticate by providing knowledge about something secret, such as entering a password. For possession based authentication, users provides something only they have (often called a "token"), such as when using a specific key or an access card. When using inherence based authentication, users authenticate by providing information about something they are, such as biometric information (e. g. fingerprint, iris grain, DNA), or with implicitly derived factors, such as certain behavioral pattern (e. g. within gait or keyboard usage) which do not involve secret knowledge.

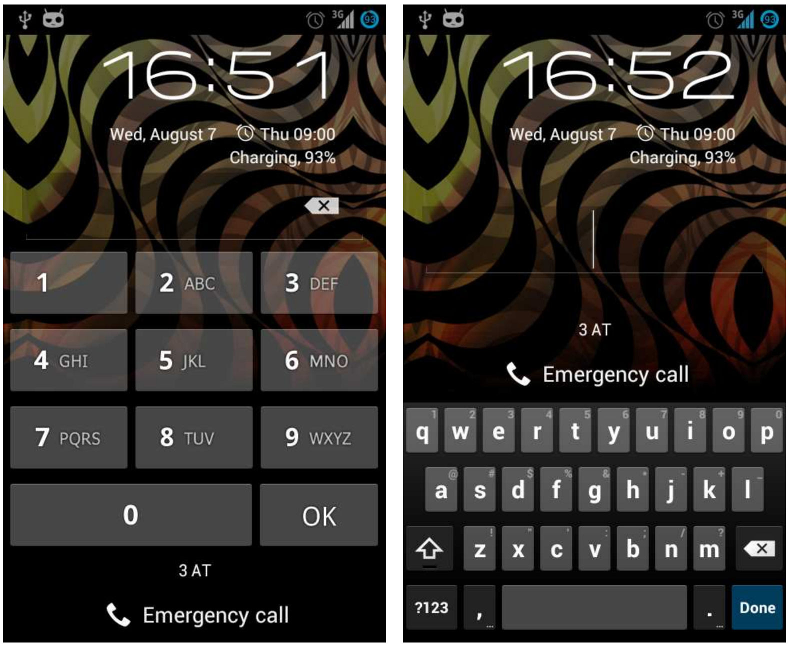
When focusing on knowledge as the primary factor in current mobile device authentication approaches (as all three, currently widely used unlocking mechanisms – PIN, password an unlock pattern – are based on knowledge), there is the problem of "cognitive load". Conceptually, users should choose different shared secrets for authentication on all devices they use – so that leaking the secret for authenticating with one device does not necessarily break authentication for all the other devices too. Therefore, the more devices users owns, the more shared secrets they have to memorize and remember. This phenomenon is often referred to as increasing cognitive load. The bigger the cognitive load of an approach is, the less is its usability – which is a conceptual problem of knowledge based authentication when used on many devices. Further, the more shared secrets users have to memorize, the less likely they will choose long, complex and hard to remember secrets. Consequently, many users either choose short and easy memorizable secrets, or often reuse a single, more complex secret. An (in most cases) accelerated authentication process makes short secrets even more attractive. All of the mentioned cases make it easier for an attacker to possibly derive the one shared secret in use, or reuse a secret leaked once for other services.

2.1 Classical Authentication on Mobile Devices

When only looking at current smartphones with activated device lock, there exists a group of three most widely applied locking mechanisms: PIN, password and unlock pattern. All of them are knowledge based, so that the user authenticates by providing information about the shared secret.

2.1.1 PIN

With a PIN based mobile device authentication the user enters a – typically 4 digit – number in order to unlock the device before usage (see figure 2.1). With using a 4 digit PIN, the key space size is 10000, resulting in an entropy of ~13.3bit. Therefore this approach could be brute forced, which can practically be rendered ineffective e. g. by an increased delay between tries. In terms of cognitive load and delay when unlocking the phone, PIN based device unlock requires the user to remember the PIN and from anecdotal experience usually takes around 2 seconds to unlock the device – which makes it significantly faster over a password based unlocking mechanism.



(a) Unlock PIN **(b)** Unlock password

Figure 2.1: Mobile device lock screen awaiting a) a PIN entry and b) a password entry for unlocking the device.

2.1.2 Password

Password based device unlock uses essentially the same approach as PIN based device unlock, with a password instead of a PIN (see figure 2.1). Before accessing the device the user enters a password, which may be of an arbitrary length and consist of letters, numbers and symbols. Assuming 80 possibilities per character[1], for a 6 character password the key space size is $2.62 \cdot 10^8$, resulting in an entropy of ~37.93bit; for an 8 character password the key space size is $1.68 \cdot 10^{15}$ with an entropy of ~50.56bit. Compared to PIN based unlocking, password based unlocking takes longer for several reasons: first, when entering a PIN only buttons for the numbers 0-9 are required, therefore keyboard buttons are usually bigger and easier to hit. Second, the majority of current smartphone keyboards is separated into several parts which each showing a group of characters belonging together, such as letters or symbols. In order to fully exhaust the entropy of a password by using letters, numbers and symbols in a mixed way, the user is required to switch in between the different parts of the keyboard. This a) makes entering a password slower than entering a PIN and b) practically prevents the majority of users to fully exhaust the entropy of password based device unlock.

2.1.3 Unlock Pattern

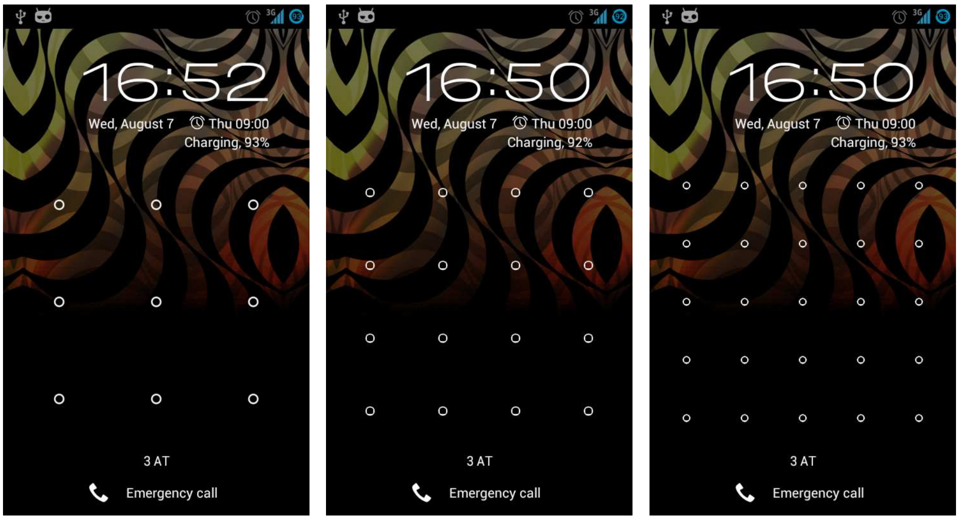
With a pattern based unlock approach users connect an arbitrary, previously defined amount of position-fixed dots on the screen of their mobile device in an arbitrary, previously defined order. Only if all defined dots have been connected in the defined order, the mobile device unlocks for usage. A pattern composed out of 9 dots in square formation is used most widely – but there exist other patterns too, which are composed out of more dots or have a different geometrical formation (see figure 2.2).

Assuming a pattern composed out of $N = 9$ dots and assuming dots can be connected in arbitrary order and length l within minimal length $l_{min} = 1$ and maximal length $l_{max} = N$, the key space size is $k = 986409$ (see equation 2.1), which results in an entropy of ~19.91bit[2]. Compared to PIN and password based unlocking approaches, using a pattern is about as fast as using a PIN – with the user having to remember the combination of dots instead of a PIN.

$$k = \sum_{l=l_{min}}^{l_{max}} \frac{N!}{(N-l)!} \tag{2.1}$$

[1] For Android the actual amount of possibilities per character varies among different builds and versions and usually is even higher than 80 characters.

[2] With Android only unlocking patterns are allowed which a) consist of 4 or more dots and b) do not create a connection between dots over other, yet unconnected dots.



(a) 9 dots **(b)** 16 dots **(c)** 25 dots

Figure 2.2: Mobile device unlock pattern using different amounts of connectible dots.

2.2 Attacking Classical Authentication for Mobile Devices

There exist many different attacks to mobile device authentication based on PIN, password an pattern unlock. The most important is the shoulder surfing attack, which is relevant to other domains too – but there also exist other attacks, such as the exemplary stated smudge and acceleration sensor attack.

2.2.1 Shoulder Surfing Attack

All three mentioned mobile device unlocking mechanisms – PIN, password and unlock pattern – are prone to the shoulder surfing attack [161, 176]. With the shoulder surfing attack, an attacker watches the display while the legitimate user authenticates, and thereby observes the shared secret. The shorter the secret used for authentication is, the easier and more inconspicuous a shoulder surfing attack can be conducted. Shoulder surfing attacks are a widely known problem not only for mobile devices, but e. g. also for entering a PIN at an ATM – which essentially is the same problem. Different approaches specifically developed to be shoulder surfing resistant have been proposed, e. g. by De Luca et. al. for ATMs by using a color scheme [53]. For the mobile domain, De Luca et. al. propose different approaches for shoulder surfing resistant authentication, such as a back-of-device authentication [54] or with implicit features derived from performing a pattern unlock [55]. Further, there exist a wide variety of graphical password schemes (e. g. [193]), for which overviews are provided e. g. by Bidde et. al. [16] or Hafiz et. al. [85].

2.2.2 Smudge Attack

Besides being prone to the shoulder surfing attack, the pattern based unlock approach is prone to another attack more specific to this approach: the smudge attack [9, 202]. With the smudge attack, attackers analyzes the display of the mobile device after the legitimate user authenticated. They thereby observe the pattern that remains on the display of the device, due to the residual grease left by unclothed fingers (see figure 2.3). Afterwards, the attackers can use a simple replay attack and use the just observed secret to authenticate with the device.

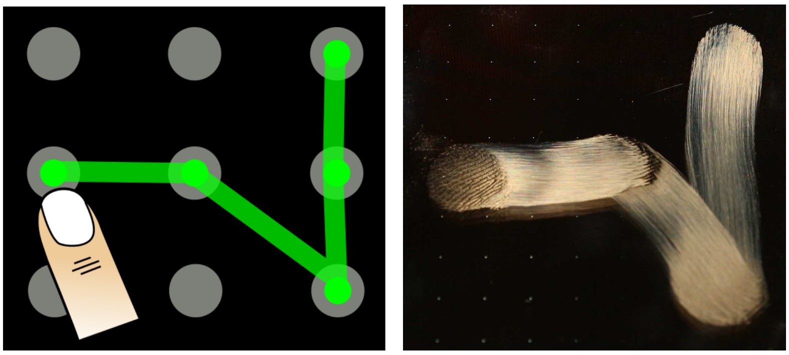

(a) Unlock pattern **(b)** Residual grease

Figure 2.3: Residual greases on the device's display after performing a pattern based unlock [202].

2.2.3 Motion Based Keystroke Inference Attack

As with the shoulder surfing attack, all three widely used authentication approaches – PIN, password and unlock pattern – are prone to the "motion based keystroke inference attack" [35] (acceleration sensor attack) under certain circumstances. For reasons of tactile feedback entering a character or connecting a dot during unlocking widely causes the device to vibrate. Assuming that vibration based tactile feedback is enabled during unlocking the device, an application running on the device could use built in acceleration sensors to record the unlocking vibrations. Based on recorded vibrations the secret used during unlock can possibly be derived, as conducted e. g. by

Aviv et. al. [8] or Cai et. al. [35, 36]. This attack requires an application to be running on the mobile device, which monitors the device's acceleration values during authentication. As this is a form of malicious software conducting a side channel attack in order to obtain a secret, the device itself has to be thought of being compromised – which distinguishes this attack from the shoulder surfing and smudge attack. The main issue with this attack is the acceleration sensor values not being thought of deserving protection at this point in time.

2.3 Biometric Authentication on Mobile Devices

Besides using PIN, password and pattern based unlock, there exist a vast variety of other authentication approaches for mobile device, such as using context [169], NFC tags or image based gesture puzzles [163] as authentication and access criteria for mobile devices. Another concept for mobile device authentication is using biometric information, which is a form of inherence based authentication: users authenticate by providing information about something they are. Consequently, biometric authentication is conceptually resistant to the shoulder surfing attack. Typical steps with biometric authentication are:

1. Obtain input data, containing the user's biometric information.
2. Extract or derive features from the obtained data. This features have to be discriminating amongst different users.
3. Recognize the user based on the extracted features. This is often done using a distance measurement between features and/or by using a learning approach.

Widely known forms of biometric information used for authentication include using fingerprint, DNA, retina and face [52, 99]. Besides those, there exist other approaches, such as hand-, gait-, ear-, voice- or even shaking-based recognition (e. g. [141]). When using biometric authentication in the mobile domain, additional hardware is required for some approaches by now. The most important approaches conceptually applicable in the mobile domain with current device technology are described in detail below.

With every approach to directly using biometric information for authentication, key revoke can be prohibitively difficult (i.e. when the stored template or reference images were compromised and the authentication data would therefore need to be changed). Consequently, authentication based on biometric information should not target high security systems, but – as example for our approach – personal mobile devices that are in frequent use, where this approach is more convenient to use and still provides a higher security level than current approaches.

2.3.1 Speaker Recognition

The idea with using speaker recognition is recognizing users by their voice. First, an audio stream is recorded with the user speaking either a predefined or a randomly chosen text. Sometimes this stream is filtered in order to suppress noise and background voices. Then, features are derived from the audio stream, which – as for other biometric authentication approaches – are required to be distinctive. Usually, deriving distinctive features is harder with a randomly chosen text than with using a predefined text. Finally, the derived features work as input to classifiers, which distinguish between users. In terms of cognitive load the user only has to remember a key phrase with using a predefined text. In order to perform speaker recognition on a mobile device, the device only needs to contain a microphone capable of adequately recording human voice – which is basically present in each current smart phone. An approach for attacking speaker recognition is by using a replay attack (e. g. [187]), based on a recording of the legitimate user speaking either the predefined or a random pass phrase. In order to resist this attack, the system can require the user to speak a displayed text – randomly chosen by the system (e. g. [10]). This requires the approach to verify that the spoken text matches the displayed text, additionally to perform speaker recognition.

Kinnunen and Li [107] give an overview of state of the art approaches to text independent speaker recognition, Lawson et. al. [113] give an overview of state of the art approaches to speaker recognition for the mobile domain. Fatima and Zheng [61] focus on approaches to short utterance speaker recognition (SUSR), which is speaker recognition based on a small amount of training and test data. An additional challenge – specially in the mobile domain – is treating background noise present additionally to the speaker's voice [122, 131]. E. g. Rao et. al. [148] focus on noise robustness in their mobile device speaker recognition approach. They utilize multi-SNR and multi-environment speaker models consisting of neural networks for speaker recognition and evaluate their approach by adding different types and levels of noise. Chetty and Wagner [43] propose a robust speaker recognition system which is based on fusion of audio-lip motion recognition, audio-lip-correlation and 2D/3D motion range information within recognition cascades. Hautamäki et. al. [87] use maximum a posteriori vector quantization (VQ-MAP) as a simpler version of maximum a posteriori adapted Guassian mixture models (GMM-MAP) for speaker verification.

2.3.2 Gait Recognition

The idea behind gait recognition (e. g. [133–135, 183, 192]) is to distinguish users by information derived from their gait, which is their distinctive style of walking (see figure 2.4).

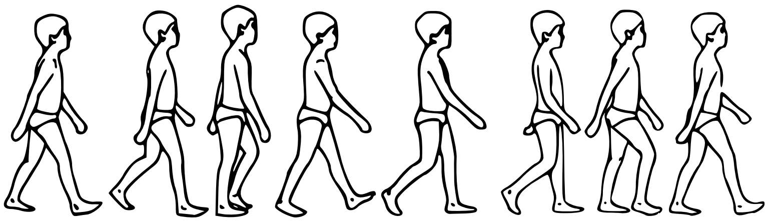


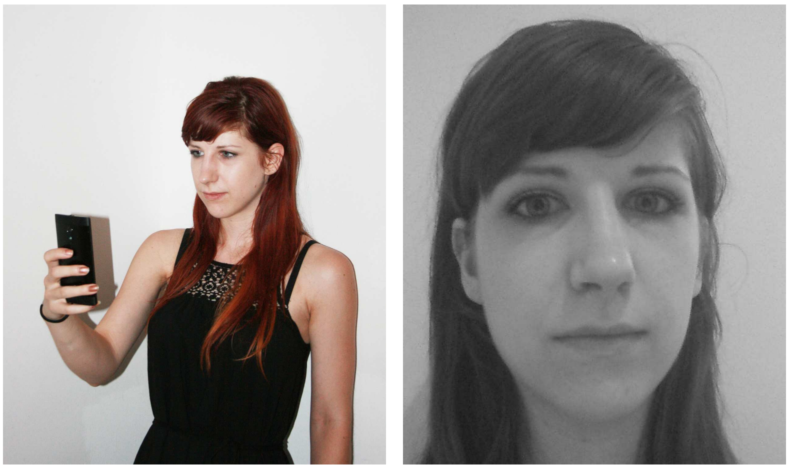
Figure 2.4: Example of gait consisting of different phases – which are used when deriving features for gait recognition [183].

First, most approaches use sensors (such as accelerometers or a gyroscope) to record the gait while the device is e. g. in a trousers pocket. This recording often is filtered to discard noise. Then, features are derived from the gait recordings. This may include an initial extraction of gait cycles (a cycle is a reoccurring unit containing two steps). Finally, as with other approaches, these features are handed to classifiers in order to perform user recognition. Gait recognition is conceptually different from the other stated biometric recognition approaches, as it is not done at a certain point in time, but continuously. With gait recognition, a user cannot instantly perform a device unlock, as gait recognition requires a) the device to be e. g. in the trousers pocket and b) a gait recording while the user walks, which is longer than e. g. the recording required for speaker recognition. Therefore, gait unlock is an implicit (passive) mobile device unlock and works as follows: users walk with the device being e. g. in their trousers pockets, and the device knows it's with an authorized user. When users wants to use their device while walking or a few seconds after walking, they can pick the device from their pocket and use it right away – as the device knows, that it has been with an authorized user up to the last seconds and assumes that its current user is a legitimate user. Consequently, the device notices and locks itself when users take the device out of their pocket and put it somewhere else.

As with all implicit authentication approaches, with gait unlock the user does not have to remember an unlocking secret, therefore does not to have to remember any cognitive load. A possibility to attack gait based recognition is by using a replay attack, which simulates the legitimate user's gait. Aggregating data for performing a replay attack is conceptually more complicated than with other approaches – as gait data is not available to the public (in contrast to e. g. data for face unlock, for which images can possibly be fetched from social networks) and cannot be recorded uncomplicated (as with voice unlock, for which data could be recorded while talking to the legitimate user). Recording a user's gait would require e. g. a malicious mobile device application installed on the user's device, which secretly records the user's gait. We are not considering this threat in further detail, as the device itself has to be thought of being compromised at this point. Besides the mentioned advantages of gait unlock, the main disadvantage is the user not being able to perform the unlock immediately at a certain point in time. Therefore, gait unlock is no alternative to the other biometric mobile device unlock approaches, but an addition in order to increase unlocking usability.

2.3.3 Face Recognition / Face Unlock

With face unlock, the mobile device unlocks for authorized users by recognizing their face, observed by a built-in camera. The core component of face unlock therefore is face recognition, which is used to distinguish between different people by their biometric facial information. First, the device records the user's face (e. g. a single photo, a photo series or a video, see figure 2.5) with a device integrated camera. Next, face detection and segmentation are used to a) find the face position in the recorded images and b) extract the face from the image to a smaller image only showing the face (e. g. rectangular crop area). Finally, face recognition is performed on extracted faces in order to distinguish between users.



(a) (b)

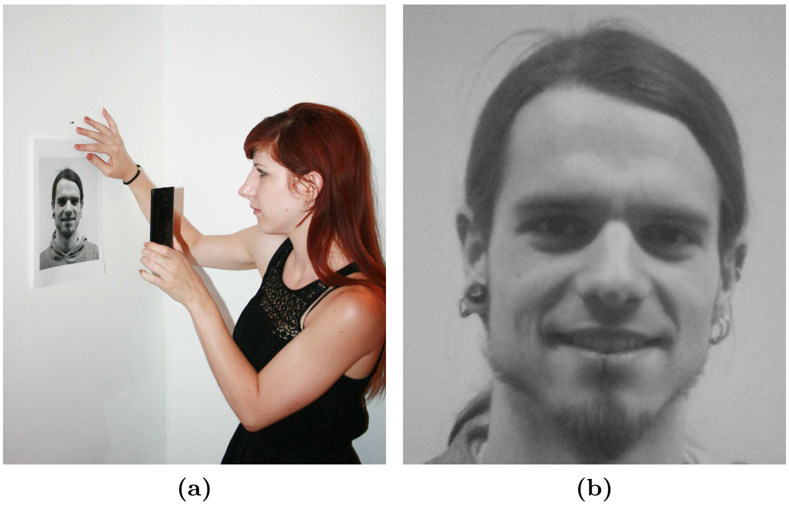
Figure 2.5: A user performing a face unlock with a) the user presenting his or her face to the camera and b) the camera recorded face image.

In terms of duration and usability, face unlock can conceptually be faster than the classical authentication approaches (PIN, password, unlock pattern) and other presented biometric authentication approaches (speaker and gait recognition). As with the other biometric authentication approaches, face unlock is not prone to shoulder surfing or similar attacks, and the user does not have to remember an unlocking secret with face unlock.

Besides these advantages, face unlock approaches are conceptually prone to the shoulder surfing attack, with which an attacker spoofs the authentication by presenting a photo or video of the legitimate user to the camera (see section 2.3.4). Only using frontal perspective biometric facial information for face unlock – which is the case for most of the currently existing face unlock approaches – makes performing photo attacks even easier.

2.3.4 Photo Attack

With a photo attack, an attacker spoofs face based authentication by presenting a sufficiently large and high-quality photo, series of photos or video to the camera (see figure 2.6). Most current mobile device face unlocking systems only utilize frontal perspective face information, which makes performing a photo attack even easier – as only frontal and no profile perspective face images have to be aggregated previously to the attack. For many people, this data can be grabbed from social networks or video platforms without restrictions and costs – as this data only yet starts being considered as deserving protection. Additionally, the grabbed data might likely be of higher quality than the data actually recorded with a mobile device by legitimate users in certain situations – in which face unlock is expected to work accurately nevertheless (e. g. legitimate users recording their faces from slightly below with additional backlight, which results in bad illumination of the face).



(a)

(b)

Figure 2.6: A user performing a photo attack to circumvent a mobile device face unlock with a) the user presenting a printed photo of the legitimate user to the camera and b) the camera recorded face image.

There exist different approaches specially developed to prevent photo attacks in face authentication approaches (overview e. g. Pan et. al. [139]), with an excerpt being presented here. Wagner and Chetty [188] provide an overview of state of the art liveness assurance approaches for face authentication systems to overcome photo attacks. A common approach to liveness assurance is eye blinking, such as used in combination with pupil movement by Teja [177]. The face authentication system of Frischholz and Werner [75] instructs the user to look into certain directions during authentication. Using head pose estimation, the system then recognizes if the user reacts according to the instructions. Tronci et. al. [179] combine video and static frame analysis of faces to avoid photo attacks. Bao et. al. [11] use an optical flow field to determine if the recorded face is on a two dimensional plane instead of being a three dimensional head. With not only including visual information during authentication, Bredin et. al. [27] propose an approach based on face and speech authentication, which aims to be replay attack resistant by approving the correspondence between audio and visual information recorded during authentication. Bharadwaj et. al. [15] use motion magnification for facial spoofing detection in videos. They detect and enhance small facial expressions in order to detect local binary pattern texture features. They further perform motion estimation using HOOF optical flow descriptors.

In order to test capabilities of resistance to photo attacks, there exist several photo attack databases containing photos and videos for spoofing attacks, such as the Print-Attack database by Anjos and Marcel [7] or the Replay-Attack database by Chingovska et. al. [44].

Chapter 3

Related Work

In this chapter we present a comprehensive review of face unlock approaches and their most frequently used core components. Most existing authentication approaches based on biometric face information conceptually feature a face detection, face segmentation and face recognition module (see figure 3.1).

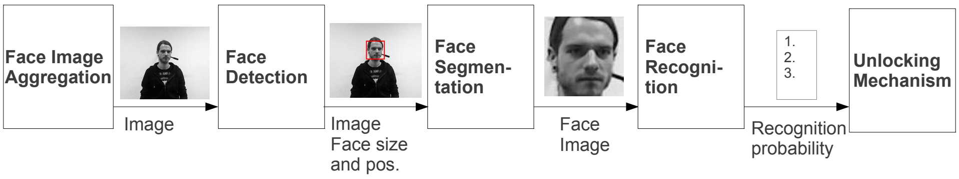


Figure 3.1: Face detection, segmentation and recognition as frequently used core components of a face unlock toolchain.

The first module (face detection) is used to localize faces in recorded images. For mobile device unlock based on biometric face information, there is only one face to find (face localization) in the regular cases. The second module (face segmentation) extracts faces localized by the face detection module from recorded images to separated, smaller images. In most cases, the face segmentation module is integrated into the face detection module and not mentioned separately, as it is very simple (such as cropping the image to the rectangular area the face was found in). The final module (face recognition) checks the user's identity based on the segmented face images in order to decide on authentication.

Manabe et. al. [123] provide an overview of biometric authentication approaches on mobile devices. Tao and Veldhuis [175] propose a mobile device face unlock approach using Haar-like feature based face detection, a local binary pattern based filter to achieve illumination invariance and likelihood ratio feature verification for face recognition. They evaluate their approach with photos recorded with mobile devices and using the Yale Face Database B [80]. Abdel-Hakim and EI-Saban [2] implement a mobile face authentication system using a graph model for face representation and low rank matrices composed of the graph attributes with Euclidean distance measurements for face recognition. They evaluate their approach on a small dataset recorded with mobile device cameras and using the FRGC face database 2.0 dataset [145]. Ijiri et. al. [98] implement and evaluate an face unlock system for mobile devices using studio photographs – although they don't describe their test data or the used face detection and recognition approach in detail. Chen et. al. [42] describe a multi-user face unlock approach based on sparse coding (requiring less samples) that they mention to be applicable to the mobile domain. They use Eigenfaces and a k-nearst neighbor algorithm for recognition, but don't describe their face detection and segmentation approach.

Beside these approaches, there exist many hybrid authentication approaches designed for usage in the mobile domain. Most of them have been implemented and evaluated for research purposes, but not yet been implemented and made available for broad usage on mobile devices. A face and eye detection for mobile devices is developed by Hadid et. al. [84], based on Haar-like features and AdaBoost. They verify their approach by using local binary patterns for face recognition and authentication. In recent research, McCool et. al. [128] report increased authentication rates by combining real-time face and speaker recognition for mobile device authentication in the MoBio project. So do Tresadern et. al. [178], again in the MoBio project – they localize a face in size and position using sliding window face detection and cascaded local binary pattern classifiers. For face normalization, they fit the face shape and texture using active appearance models, then remove background information and transform the face to a normalized shape with standard brightness and contrast. For face recognition, they first remove illumination effects using gamma correction, difference of Gaussian filtering and variance equalization. Then they compute three differently sized local binary patterns for every pixel and use the resulting histogram as feature vector for classifiers, for which they use simple distance measurements. Similar approaches have been implemented and evaluated, e. g. by Mayrhofer and Kaiser [127] and Shen et. al. [170], who also report improved authentication results with fusing face and speaker recognition in their mobile device authentication approach. Kim et. al. [106] extend the fusion of face and speaker recognition by using teeth recognition in their multimodal authentication approach for mobile devices.

3.1 Face Detection and Face Segmentation

Face detection is finding human faces in images, if there are such. This task most commonly includes finding the position and size of the face, but may also include finding the rotation and perspective of the face. Face segmentation deals with the extraction of faces found by face detection from the originally recorded images. Separating face and non-face related information in images is an important prerequisite e. g. to face recognition, which conceptually should only utilize face related information. In many cases, face detection and face segmentation are performed together in a single step or directly one after another – and face segmentation is not mentioned as a self-contained component. Both are commonly used as prerequisite to face recognition, but also in advertisements or with autofocusing on faces with digital cameras. There exist many concepts to face detection and segmentation. In general, these approaches can be grouped into two top level classes, such as done by Hjelmås and Low [92] (see figure 3.2): biometric/geometric feature-based and image-based (view-based) face detection approaches.

3.1.1 Face detection with biometric/geometric features

Face detection based on biometric/geometric facial features uses knowledge about the alignment of a human face elements, such as position of eyes, nose, mouth, ears and eyebrows, the face contour or brighter/darker skin areas caused by shadows of the face surface structure. As these approaches need face related features in order to find a face by design, they conceptually cannot be applied to problems other than face detection without major modifications. Further, when detecting faces from different perspectives, likely different biometric features have to be derived – which results in structural different face detectors for different perspectives. Hjelmås and Low [92] group face detection approaches based on biometric/geometric facial features in three further classes: low-level analysis, feature analysis and active shape models. Low-level analysis based face detection derives visual features from the image pixels. This include edges, differentiation between grayscale and color pixels or – if a video is available – changes of pixels between frames. The problem of features derived by low-level analysis tending to be ambiguous is addressed by feature analysis based face detection. There, a high-level feature analysis is performed on features derived by low-level analysis in order to verify them, either by checking their constellation or by deriving features in a predefined order based on previous knowledge. With active shape models, the knowledge about facial-feature constellations is used to form a shape model, which then actively tries to match a potential face in an image. Amongst the important approaches which evolve towards a potential face location are snakes, deformable templates and point distribution models.

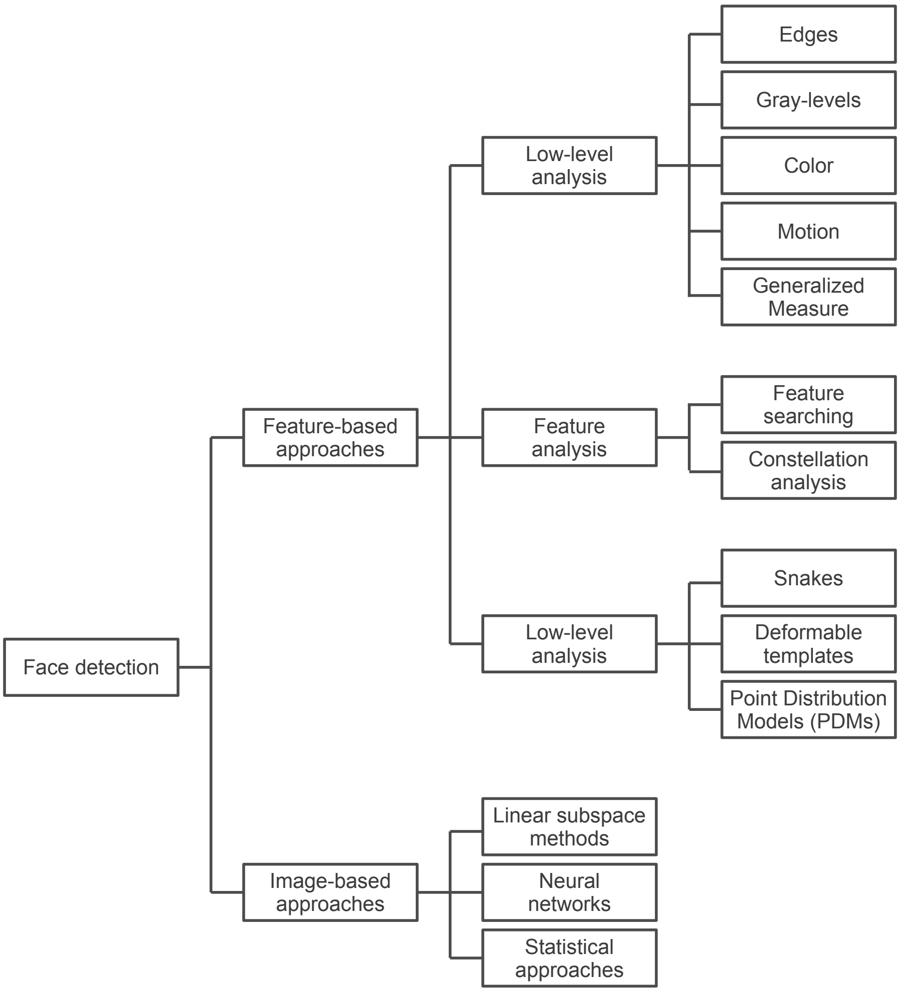


Figure 3.2: Classification of face detection approaches by Hjelmås and Low [92].

3.1.2 View-based face detection

As deriving biometric/geometric facial features explicitly from prior knowledge is error prone to many different external influences (such as changes in rotation, image illumination or background information) there exist approaches deriving these features implicitly within view-based face detection. View-based face detection approaches use image pixels for detection without making use of biometric and geometric facial features explicitly. Therefore, these approaches usually require training data, which acts as prior knowledge and from which features are derived implicitly. Consequently – in contrast to feature based face detection – view-based approaches are conceptually applicable to face detection from different perspectives and even non-face related detections without major changes to the approach (usually only different training data is required). In their survey, Hjelmås and Low [92] group view-based approaches into the following three groups: approaches based on linear subspace transformations, on neural networks and on statistical approaches. Linear subspace based face detection aims at transforming the face into a face space – other dimensions better representing faces. Among these approaches are e. g. the well known principal component analysis (PCA) and linear discriminant analysis (LDA) [124]. Neural network based approaches learn discriminating facial features implicitly from training data and frequently include transformations and/or filtering of pixel values as preprocessing. Statistical approaches to face detection include e. g. using support vector machines and decision trees/decision networks.

3.1.3 The Sliding Window Principle

Especially with using view-based face detection approaches a commonly used technique is the sliding window principle. With sliding window face detection, a search window smaller than the original image is shifted through the image with an arbitrary stepwidth (see figure 3.3).

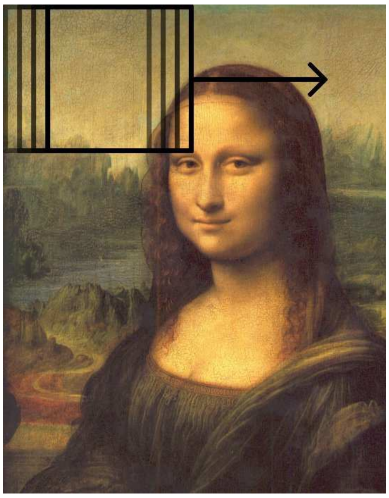


Figure 3.3: With sliding window face detection, a search window is shifted through the image inside which face detection is performed [64].

On each position the search window is shifted to, face detection is performed on the part of the image currently contained in the search window. In order to find faces of different sizes with a sliding window principle, more than one such sliding window processes with differently sized search windows are used. At a given search window size and position, the face detection has to decide if the current window actually contains a face or not (e. g. with using a probability value and a threshold separating between face and non-face detections). As there will be multiple detections with slightly changed search window positions and sizes next to each other, only the match with the highest probability inside a certain area will be accepted as detected face. Further, some sliding window face detection approaches include multiple phase detection: in the first phase, a coarse stepwidth is used for shifting the search window across the image and for scaling the search window. In later phases, these stepwidths are decreased in order to match a face position and size more precisely. On each position and size of the search window, face detection is performed on the image part currently covered by the window.

3.1.4 Challenges of Face Detection in the Mobile Domain

In literature, face detection and segmentation are often considered to be widely solved problems due to the vast amount of approaches delivering promising results. This assumption is based on certain further assumptions – limitations and restrictions to the test scenario – most of the approaches have been evaluated on. When speaking about evaluation data these approaches have been tested on, in many cases one or more of the following limitations apply to the test data recording scenario:

- Using roughly equal illumination conditions, especially only using a limited amount of background lightning.
- Using data with fixed, minimum image quality, e. g. only showing a limited amount of fuzziness.
- Using limited, homogeneous or roughly equal background information across all data.
- Limiting the allowed changes in participants' style and appearance, such as changed beard style, using/not using glasses or different facial expressions.
- Limiting the distance and rotation variance of the user's head.

When applying face detection in the mobile domain, these assumptions do not hold as they do for many test sets. Consequently, face detection cannot be assumed to be a widely solved problem for all scenarios. Further, mobile devices still feature less computational power than is available on most personal, non-mobile computers. Therefore, for mobile face detection approaches a certain processing speed is mandatory.

3.1.5 Face Detection in Literature

There exist a vast amount of approaches to face detection and segmentation. Important research towards both successful face detection and face recognition based on Eigenfaces was conducted by Turk and Pentland [181].

Rowley et. al. [153] use neural networks for face detection. They at first build up an input image pyramid by scaling the input image to multiple sizes, then perform sliding window based face detection. Next, they extract the image content for each window position, perform illumination correction and histogram equalization. Then, they use the extracted pixels as input to a feed forward neural network. To avoid multiple detections close to each other they finally merge overlapping detections. In succeeding research Rowley et. al. [152] extend their approach by incorporating rotation invariance. Haar-like features based on wavelet representations of objects were used by Papageorgiou et. al. [140] for general object detection and later used by Viola and Jones [186] for face detection in their well known object detection framework. Lienhart and Maydt [115] extended the approach proposed by Viola and Jones with easily rotating features to a computationally fast and de facto standard approach of face detection [185]. Sung and Poggio [174] used view-based model clusters that distinguish between "face" and "non-face" incorporating a Mahalanobis distance measurement. They evaluate their approach only on frontal face images, but the approach could conceptually also be trained for any other perspective. Bayesian discriminating features were used by Liu [119], which compare likelihood density estimations of an image to decide if an image contains a face. Schneiderman and Kanade [164–166] propose an object detector also applicable to face detection. Their approach is based on statistics of image parts extracted by using wavelet transformation. Jesorsky et. al. [101] use a face shape comparison in order to detect faces in images. The approach aims to be robust to changes in illumination and background with extracting edges from faces and comparing them by using the Hausdorff distance [154]. Kienzle et. al. [105] propose computationally fast approximations to support vector decision functions for usage in face detection. They replace derived support vectors by a smaller amount of synthesized input space points in order to reduce computational complexity. Sahoolizadeh et. al. [155] combine Gabor wavelets and neural networks for face detection and recognition. Douxchamps and Campbell [58] combine Viola and Jones based face detection with various filters to obtain a good face detection tracking rate in videos [58]. Abiantun and Savvides [3] use Real AdaBoost with 3 explicit bins (positive, negative, abstain) to obtain a single, strong face detector [3]. Dalal and Triggs [50] focus on deriving robust features for human detection – but their approach can conceptually be applied for facial detection as well. They propose the usage of grids of Histograms of Oriented Gradient (HOG) for constructing feature sets and show that their approach outperforms many previous approaches to human detection in terms of computation complexity as well as detection accuracy.

Finally, the use of skin color for face detection was investigated by different authors, e. g. Hsu et. al. [95], Martinkauppi [125] and Zarit et. al. [199], but turned out to be less reliable than other approaches.

For a more comprehensive review of existing face detection approaches we refer to the surveys of Hjelmås and Low [92], Degtyarev and Seredin [51] and Yang [198]. Further, Huang et. al. [96] review local binary patterns for facial image analysis, namely face detection, facial expression analysis and face recognition, and Santana et. al. [157] provide an overview of facial feature detectors build upon the Viola and Jones object detection framework.

3.2 Face Recognition

Face recognition is deriving the identity of people from their faces. This is done by assigning a label (identity) to face with yet unknown identity, which makes face recognition a classical pattern recognition problem. Face verification is a binary form of face recognition: it does not derive the identity from a person's face, but either confirms or negates a proposed identity for a given face. Nevertheless, face verification is often called binary face recognition or simply face recognition in literature. Face recognition is used in a wide variety of application areas, such as in surveillance (e. g. CCTV), in access controls (e. g. building/device access, border controls), in the advertising domain, in human computer interaction (HCI) or robotics. Also mobile device face unlock incorporates face recognition as a key component. As face recognition requires face information as input, reliable face detection is the most frequent prerequisite to face recognition.

As with face detection, there exist two top level approaches to face recognition: using geometric features and view-based (appearance-based) face recognition. Geometric feature based face recognition incorporates the knowledge about geometric alignment of human face elements, such as eyes, nose, mouth, eyebrows and ears or the face contour. From this geometric alignment biometric features are derived, which further are used for face recognition. In general these approaches therefore cannot not be applied to data other than faces without major modifications. Further, face recognition from different perspectives likely requires deriving different geometric features. With view-based face recognition, the pixel values itself are used for face recognition without deriving geometric features first – but may include arbitrary transformations of pixel values without knowledge about geometric features (e. g. subspace transformations such as PCA). As these approaches do not incorporate finding biometric facial elements based on prior knowledge of a face's structure (but derive implicit biometric features from training samples) these approaches can be applied to different perspectives and even data other than faces (again requiring corresponding training data).

3.2.1 Face Recognition Accuracy Measurements

When looking at the task of successfully performing face recognition, face detection must provide good results as input to the recognition in terms of a) a high rate of correct detections and b) a good face normalization. When looking at normalization, it is important that all detected faces roughly include the same area of facial information (e. g. from the left to the right ear and from the hair line to the chin) – and that inside the area marking a face, the faces should be positioned equally (such as centered at the nose). When measuring the face detection rate itself, the performance can be stated as amount of correct detections (true positives) and the amount of wrong and missing detections (false positives and false negatives). A low true positive rate means that many faces are missed during detection – which causes the face recognition to have less data available during training and classification. A high false positive rate means that many detections don't actually contain faces – which causes the face classifiers to learn from non-face images. Both cases will decrease the recognition rate and should therefore be avoided.

Unfortunately, the detection rate is not the only factor influencing face recognition. Estimating the detection rate for a specific face detection approach depends on making a binary choice for each of the images if the face was detected correctly or not. This includes a tolerance in terms of normalization so that faces e. g. slightly shifted to one side, scaled slightly differently or with a certain amount of background information still present will also be counted as correctly detected faces (see figure 3.4).

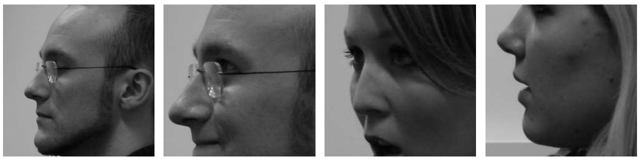


Figure 3.4: Face images after face detection showing background information and unequal normalization in size and position [62].

If the grade of face normalization provided by face detection and segmentation is not sufficient, subsequently applied face classifiers will not only learn the face-discriminating features, but also discriminating features in normalization[1]. E. g. if the face of subject A is shifted to one side of an image, a classifier will also learn the shift besides learning the face properties. If a face of subject B has the same shift, the classifier will more likely classify this face as originated by subject A.The same applies for background information present in face images after face detection, e. g. with using a rectangular crop area for face segmentation. Again, face classifiers will learn discriminating

[1] When not learning from geometric but appearance-based features.

features in background information additionally to the face-discriminating features. Consequently, the detection rate itself is a poor indicator for the impact of face detection quality on the subsequent face recognition step.

3.2.2 Challenges of Face Recognition in the Mobile Domain

In general face recognition is – in contrast to face detection – not yet assumed to be a widely solved problem. One of the reasons for this conclusion is that face recognition relies on feasible and normalized face detection results, which is a complicated task itself when not incorporating scene restricting assumptions. Further, face recognition faces the same challenges as face detection, which again lead to decreased correct recognition rates. Amongst the overall challenges of face recognition are (e. g. Khashman [104]):

- False face detection results are passed to face recognition as input. This leads to the face recognition learning non-face related information.
- Differently normalized face detection results handed as input to face recognition, e. g. face images with slightly changed face position and size inside the image. This can lead to the face recognition learning features related to the normalization besides learning biometric facial features.
- Changing illumination conditions and backlight. Again, this can lead to face recognition learning features not related to biometric face information.
- Bad image quality, such as small image dimensions, bad image sensor quality, motion blur or depth of field. This possibly leads to the loss of important biometric information.
- Background information still included in face images after face detection and segmentation. Especially strongly changing background information can lead to face recognition learning features related to background information.

Not all of these challenges and possible problems apply to all approaches of face recognition. E. g. an approach based on geometric facial features can possibly avoid learning from background information at all – assuming a correctly detected face.

3.2.3 Face Recognition in Literature

There exist a vast amount of approaches to face recognition, from which we are stating an excerpt here in order to present the huge diversity of face recognition approaches. The first approach to face recognition recognized widely as successful was proposed by Turk and Pentland [181], which incorporated Eigenfaces for face detection as well as face recognition. Their work is based on previous research by Sirovich and Kirby [171], which were the first to publish the usage of Eigenfaces for face representation. Besides others, they identified changes in illumination as a major problem to their approach. Belhumeur et. al. [13] address this problem with their approach of using FisherFaces for facial recognition. They evaluate their approach in direct comparison to the baseline of Eigenfaces for recognition. Additionally, Brooks and Gao [31] perform an evaluation of FisherFaces across pose. Georghiades and Belhumeur [80] also address changes in illumination as well as viewpoint. They therefore use a view-based approach with training samples showing all illuminations and and poses to automatically reconstruct face shapes. A comparison of geometrical feature based and view-based face recognition was conducted by Brunelli and Poggio [32, 33]. Gordon [82] combines images of faces recorded from frontal as well as profile perspective in order to perform face recognition. They at first normalize input images and extract different geometrical facial features. Based on this informations, they extract several parts of the images and use them for face recognition. Neural networks for face recognition are discussed by Mitchel [132]. Lin et. al. [116] propose using probabilistic decision-based neural networks (PDBNN) for face detection as well as recognition. At first they determine the size and position of a face inside an image, then locate the eyes inside the face for normalization reasons. Then, they extract regions containing eyebrows, eyes and nose and perform face recognition using their proposed classifiers. Wiskott et. al. [194] propose the usage of elastic bunch graph matching for detecting and/or recognizing faces using only a single image per subject. In order to incorporate face images with different normalization in terms of position, size, facial expression and pose, they form image graphs out of geometric facial features. They extraction of an image graph is based on a previously build bunch graph, which is positioned using elastic graph matching. They further use different graph structures for finding and recognizing faces. Li and Lu [118] propose the usage of feature lines, which connect features of the same class in an arbitrary feature space. For face recognition, they further use nearest feature line face classification. Bourel et. al. [22] propose a facial feature extraction approach and use geometric facial features to perform tracking tasks. Cootes et. al. [47] also extract feature points from faces in order to capture the shape of faces. They explicitly target the problem of geometrical facial features mainly being used for near-to-frontal perspectives and evaluate their approach with multiple viewpoints around participants' heads. Gao and Leung [78] perform face recognition based on line edge maps (LEM). They address changes in illumination as well as changes of pose and facial expressions in their approach. Further, they state LEM as a form of representing faces which might be useful for further facial processing tasks, such as facial expression recognition. Meng et. al. [130] use radial basis function based neural networks in order to train their face classifiers with only a small amount of training data in comparison to amount of features – a problem frequently encountered in face recognition. Liu and

Wechsler [120] use a Gabor-Fisher classifier (GFC) to perform robust face recognition in terms of changing illumination and facial expression. At first, they derive Gabor features from face images, then reduce the Gabor feature vector size by applying the Enhanced Fisher linear discriminant model. Local Binary Patterns were proposed by Ojala et. al. [138] for scaling and rotation invariant texture classification and later used by Ahonen et. al. [4] for face detection. They split face images into smaller subregions, extract local binary pattern histograms which they concatenate to a single feature vector and use a nearest neighbor algorithm to perform the actual face recognition. Tsalakanidou et. al. [180] use Eigenfaces to recognize faces based on color as well as range information. They focus on stating usability of range information in face recognition and evaluate the usage of separate classifiers for color and depth as well as a combining both features spaces for classifiers. Overall, they state a significant increase in recognition accuracy incorporating facial range information. Geo et. al. [77] use a fusion of multiple views of a person's face for face recognition, which is similar to our pan shot approach. They evaluate their approach using the Stirling face database (PICS)[2]. Bronstein et. al. [30] address expression variant face recognition with their approach to 3D face recognition. They map 2D facial texture images onto 3D geometry, then use PCA to derive comparative features for recognition tasks. Weyrauch et. al. [191] perform face recognition using component-based 3D morphable models. They address illumination and pose invariance: they use a 3D morphable model of a human head to create 3D models of users' heads, using only three 2D images of each user projected onto the 3D model. Based on the model, different components are extracted and used for face recognition. Riaz [149] directly compares different implementations of neural networks, Hidden Markov Models (HMM), principal component analysis (PCA) and independent component analysis (ICA) for face recognition. Venkataramani et. al. [184] compare correlation filter, individual PCA and FisherFaces as approaches to face recognition in the mobile domain. They evaluate their implementation using an image database created from mobile devices. He et. al. [91] propose Laplacianfaces for facial representation, which is based on Locality Preserving Projections (LPP) as a form of facial subspace transformation. Based on this representation arbitrary pattern recognition mechanisms can be applied to perform face classification. Nazeer et. al. [137] also use neural networks for face classification. They extract facial features from detected face images, normalize these using different approaches (incorporating e. g. histogram equalization and normalized correlation) and finally perform neural network based face recognition. Klare and Jain [109] introduce comparative measurement criteria for the effectiveness of facial features by using three levels, ordered

[2] Psychological Image Collection at Stirling (PICS), available at http://pics.psych.stir.ac.uk/

by specificness. Kurutach et. al. [112] use trace transform to obtain view-based facial features and perform face recognition based on the Hausdorff distance [154].

For a more comprehensive review of face recognition approaches we refer to the surveys of Abate et. al. [1], Akarun et. al. [6], Bowyer et. al. [23], Chang et. al. [40], Chellappa et. al. [41], Gong et. al. [81], Huang et. al. [96], Iancu et. al. [97], Jain et. al. [100], Jones [102], Kittler et. al. [108], Scheenstra et. al. [162], Wechsler [189], Zhang and Gao [204], Zhao et. al. [205] and Zou et. al. [206].

Chapter 4

Building Blocks

4.1 Range Algorithms

We utilize range images for face detection and recognition starting with the third stage of our implementation (see section 7.3), therefore review the most important approaches to algorithmically obtain range information in short. A range image essentially is an image which represents the camera to object distance (depth) in each pixel – e. g. a brighter pixel correlates to a smaller, a darker pixel value to a larger distance or vice versa. Beside others, range images are utilized in the fields of computer vision, such as scene reconstruction and object detection from neurobiology to robotics. There exist different approaches to construct range images, with the most important being structured light and stereo vision. Because of both approaches use more than one devices (either projector and camera or two cameras), calibrating these system is very important.

4.1.1 Range Information from Structured Light

Using structured light for obtaining range images conceptually works as follows (e. g. [12, 24, 38, 66, 156, 160, 203]): a projector unit projects structured light onto an arbitrary formed surface. Structured light is a known light pattern, which is easy to observe using computer vision techniques. Therefore, structured light is typically organized in lines or dots (such as for the Kinect system). Besides the projector there is a camera, which is mounted with a known relative distance and angle to the projector. The camera observes the structured light from this slightly different point of view and extracts the pattern using computer vision. As there is a distance between projector and camera, the sensed light pattern will look slightly differently from the projected pattern, depending on the surface structure. Based on this pattern and the known setup of projector and camera (relative distance and rotation in 3D), the structure reflecting the pattern can be calculated. In order to obtain the exact setup information, calibration is required. Utilizing structured light for range image recording has several advantages and disadvantages: on the one hand, the technique itself is conceptually robust and can be used without an external source of light. Using light not visible to the human eye in the projector further enables such systems to work in virtual darkness. On the other hand, structured light requires a precise projector, which is uncommon hardware for current mobile devices.

4.1.2 Range Information from Stereo Vision

Using stereo vision for obtaining range images conceptually works as follows (e. g. [26, 56, 86, 110, 114]): two cameras record the same scene from different perspectives; using the two images the 3D structure of the scene is reconstructed. The principle is one of those used intuitively by humans: to obtain two slightly different images of the same scene using two eyes in order to observe depth (stereopsis). As the two cameras also look at the same scene from slightly different perspectives, the recorded images will look slightly different too. Utilizing the exact camera setup (relative distance between cameras (eye distance) and rotation in 3D), the range for each pixel can be calculated using stereo to range algorithms. As for using structured light, the required, exact camera setup is obtained by calibrating the system. On the one hand, the main advantage of stereo vision over structured light for obtaining range images is that it requires less special hardware (stereo cameras instead of projector and camera). There already exist several mobile devices featuring stereo cameras by now. On the other hand, the approach conceptually relies on an external source of light and therefore is not as universal as structured light. As with structured light, this approach can be used in observed darkness too using an external source of light invisible to the human eye and corresponding cameras.

4.2 PCA and Eigenfaces

An early approach to successful face detection and recognition was based representing faces with Eigenfaces [171, 181] which we utilize in the first stage of our implementation (see section 7.1). With using Eigenfaces, faces basically are transformed into a subspace (face space) using principal component analysis (PCA). In this subspace, a face is represented by a weighted combination of all Eigenfaces (which are the face space dimensions). As the core component of Eigenfaces is transforming faces into the face space using PCA, we explain the concepts of PCA and Eigenfaces in detail below.

4.2.1 Principal Component Analysis (PCA)

Principal component analysis (PCA) transforms a set of samples S from their original dimensions D_O to new dimensions D_N, so that D_N shows the maximum amount of variance amongst data samples. PCA internally uses orthogonal transformation for finding the new dimensions. Therefore, the first dimension of D_N is chosen so that it shows max. variance amongst data samples. The succeeding dimensions are chosen so that they a) are orthogonal to all previously chosen dimensions of D_N and b) show maximum variance amongst data samples.

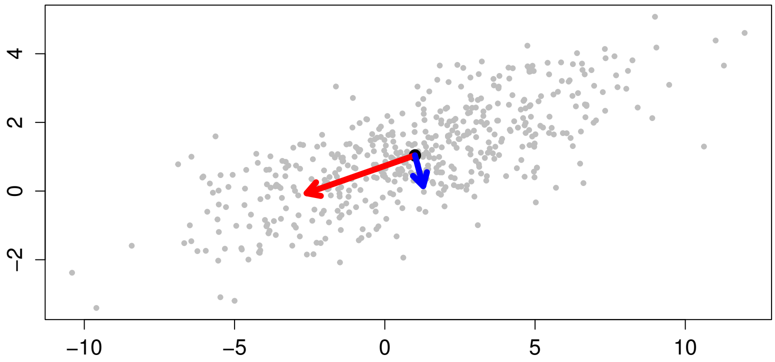
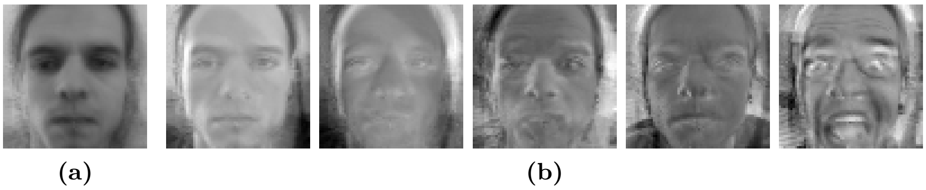


Figure 4.1: The first (red) and second (blue) principal component are derived from a 2D point cloud using PCA. The black dot in the center is the average point which acts as the point of origin for all projections to PCA derived dimensions.

Mathematically, PCA can be calculated using Eigenvalue decomposition (decompose data into Eigenvectors and their corresponding Eigenvalues). As it therefore also depends on the value scaling applied in original dimensions, it is important to normalize data before performing PCA. The amount of dimension in D_N is conceptually the same as the amount of dimension in D_O when using PCA as an exact, reversible transformation – but often the less important dimensions of D_N are left out when further processing data. This is done as a) usually samples can be represented fairly well using only the most important dimensions and b) using less features eases subsequent processing. PCA is therefore often spoken of and used as a subspace transformation, in which data is transformed to a representation requiring less dimensions than within the original feature space.

4.2.2 Eigenfaces

With using Eigenfaces in facial image processing, face images are transformed from their original dimensions D_O (image pixels) into new dimensions (face space) using PCA. The face space dimensions (Eigenfaces) are the principal components calculated by PCA using a certain amount of face images. Each Eigenface can be thought of being a scalable difference of the average face in face space towards a certain face space dimension. Faces represented in face space therefore are composed of a sum of the average face and the differently weighted Eigenfaces. For this reason, Eigenfaces themselves look similar to actual human faces when being transformed back to the original dimensions D_O (see figure 4.2).



(a) (b)

Figure 4.2: Eigenfaces look similar to human faces when transformed from face space to the original image dimensions (pixels) with a) the average face used and b) the first five, derived Eigenfaces.

With using an Eigenface representation of faces, faces can be approximated well with only using a limited amount of the most important Eigenfaces without actually losing much information. Therefore they are used within a certain amount of applications processing faces – with the most widely known approach being Eigenfaces for recognition by Turk and Pentland [181] designed for face detection as well as face recognition.

4.3 Classifiers

Starting with the second stage our our implementation (see section 7.2) we utilize the standard approaches of support vector machines and neural networks as face recognition classifiers. In the context of machine learning/-pattern recognition, a classification problem essentially is determining the class of a sample, of which the class is yet unknown – based on samples with already known classes [18, 195]. Therefore, classification is a form of supervised learning. E. g. for face recognition, a face image with unknown originator's identity is the sample to be classified. The samples with known originators' identities are the source of information, on which the classification bases its decisions. The instance performing the classification is called classifier: it essentially implements a classification algorithm, which first derives (learns) how to distinguish between classes from samples with already known classes (training data). Based on training data, a classifier is able to determine the class of a samples with yet unknown class (classification). For training and classification, samples are handed to a classifier in the form of an arbitrary amount of features (feature vector), with a features being measurable property derived from the sample. E. g. for face recognition, a face image feature vector could contain (amongst others) the image pixel values and/or numerical properties derived from the face geometry. With thinking of each features as an own dimension, each sample represents a point in a features space – which is the basis for simple classification approaches based on feature space distance measurements between samples or nearest neighbor algorithms, as well as for more complex classification approaches.

Amongst well known classification algorithms such as decision trees and Bayesian approaches, there exist neural networks and support vector machines for classification. For both, input data normalization is required, as discussed in detail in [83, 88].

4.3.1 Support Vector Machines

The approach of optimal, linear class separation on a hyperplane was originally proposed 1963 by Vapnik and Lerner [182]. Incorporating the concept of using large margins for classification [5] Boser et. al. [21] proposed the kernel trick for optimal, nonlinear class separation on hyperplanes – by transforming data into a higher dimension using a predefined transformation (kernel), in which it classes are better separable.

Support vector machines for classification/pattern recognition [34, 49] are large margin classifiers: they chose the separation between classes so that a) samples are optimally classified and b) margins between samples of different classes are maximized (see figure 4.3).

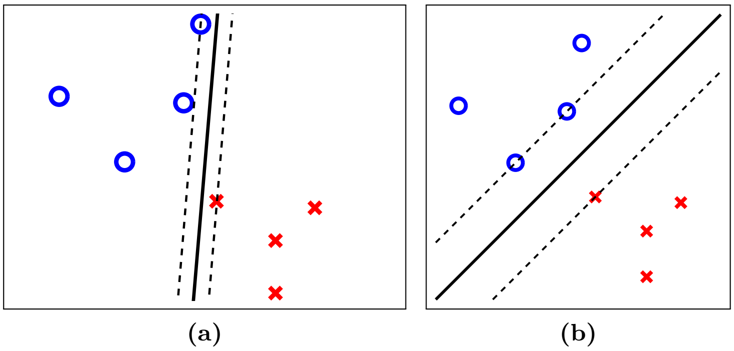


Figure 4.3: The concept of large margin classification: a) a nonoptimal and b) an optimal separation between classes in terms of maximizing the margin between samples of different classes (adapted from [59]).

A support vector machine tries to find the optimal linear separation of two classes using given samples in an arbitrary feature space, based on large margin classification. The classification is done using a linear separator between two classes (as shown in figure 4.3). Therefore, multi-class classification with support vector machines conceptually has to consist of multiple one vs. all or one vs. one classifications. Samples are classified using the shortest distance d between the sample and the class separator. For a given sample d is intended to indicate the class and distance to the separator with a) $d \geq 1$ for samples belonging to the first class (positive class) and b) $d \leq -1$ for samples belonging to the second class (negative class). Therefore, the space covered by $-1 < d < 1$ is intended to be sample free. The samples closest to the separator conceptually lie exactly on the class margin borders (dashed lines in figure 4.3), consequently have a distance of $d = -1$ respectively $d = 1$ and are called support vectors.

Using this hard, sample free margins within $-1 < d < 1$ is a form of hard margin classification – which is prone to overfitting: if outliers are included in data samples, the support vector machine will chose the separator so that outliers are also classified correctly and therefore will make the class margin smaller (or prevent data from being linearly separable). In order to prevent overfitting, hard margin classification can be relaxed to soft margin classification, with which samples are allowed to be positioned within the margin or even to be classified incorrectly – which consequently leads to $d < 1$ respectively $d > -1$ for these samples. With soft margin classification, samples lying within the class margin are called support vectors. These support vectors cause a total error E which is essentially computed as sum of support vector distances to the class margin border of their class. During the optimization process a support vector machines performs for data separation, E influences how far samples will be allowed into the margin – therefore E can further be weighted by a factor $C \geq 0$ (cost):

- A smaller C will cause a small penalty for errors caused by support vectors, therefore allow bigger errors and will lead to class separation on the one hand being less prone to overfitting, but on the other hand misclassifying a bigger amount of samples used for training.
- A bigger C will cause a big penalty for errors caused by support vectors, therefore allow only small errors and will lead to class separation on the one hand being more prone to overfitting, but on the other hand misclassifying a smaller amount of samples used for training.

Support vector machines are conceptually designed to perform a linear separation amongst classes. As real life data often is only nonlinearly separable, support vector machines incorporate the concept of transforming samples from their original dimensions D_L into higher dimensions D_H, in which a linear separation is easier. The transformation is a mathematical function $k(x) : D_L \to D_H$, which is called kernel. As finding the optimal, linear separation in D_H will computationally be more intensive due to the increased amount of dimensions, an equivalent of separating data can be computed in D_L directly (kernel trick). In order for the kernel trick to be applicable, a kernel must be admissible (the Gram matrix of the kernel must be positive finite). Amongst the most widely known and applied kernels are the linear kernel, the Gaussian/radial basis function kernel, the sigmoid kernel and the polynomial kernel. Each kernel can be configured using several individual parameters, as explained in detail in e. g. [49].

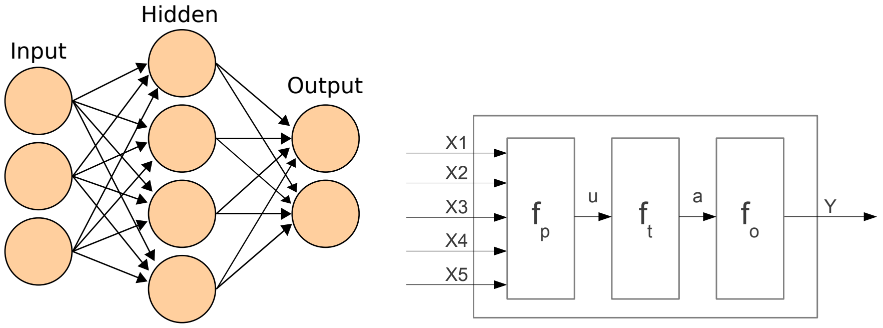
4.3.2 Neural Networks

Artificial neural networks (often called neural networks only) try to model the concepts used in a biological brain, using neurons as sources of decisions and synapses as connections between neurons. The first form of an artificial neural network was proposed 1943 by McCulloch and Pitts [129] in the form of perceptron, which originally was a networks consisting of a single neuron. Incorporating Hebb's hypothesis of neuron cooperation being correlated with their spatial distribution [89] neural networks containing multiple neurons and layers were formed – amongst them feed forward neural networks. Using multiple layers of neurons required a different approach to learning (such as [89, 93]). A widely used form of supervised learning for feed forward neural networks is with using backpropagation [88], with which measured errors are propagated back through the network in order to learn the intended behavior.

For classification/patter recognition, feed forward neural networks [17] try to iteratively learn the correlation between an input pattern (feature vector) and output pattern (classification result) from training data samples. Therefore, a pattern recognition feed forward neural network are (in a very generalized way) structured as follows[1]. They contain a) an input layer, in which neurons take the input pattern, b) an output layer in which neurons indicate the classification result and c) an arbitrary amount of hidden layers (typical one) which pipe informations (signals) between the neurons of neighboring layers. Each of the neurons in the network is connected to each neuron in the previous and successive layer (see figure 4.4a). Each of these connections holds a weight Ω (usually in the range $[0, 1]$) which is responsible for amplifying/damping signals piped over this connection. These weights usually are initialized randomly before starting the training. Each neuron contained in the network (see figure 4.4b) further contains a) a propagation function $f_p : x \to u$ which combines the weighted signals from neurons of the preceding layer to a scalar value u (e. g. sum function), b) a transfer function $f_t : u \to a$ (e. g. step function or Sigmoid function) which is responsible for the neuron firing a signal itself and c) an output function $f_o : a \to y$ which is responsible for the actual output signal form.

A neural network conceptually learns by adapting the weights of connections between neurons. In case of a pattern recognition feed forward neural network, the network learning approach tries to change the weights in a

[1] There exist a vast amount of different forms of neural networks with basically no conceptual restrictions to modification. Therefore, we only describe a standard feed forward neural network for pattern recognition at this point.



(a)

(b)

Figure 4.4: a) feed forward neural network with an input, a hidden and an output layer of neurons[2]and b) a single neuron combining input signals using an propagation function f_p, deciding on if the neuron fires a signal itself with a transfer function f_t and deriving an output signal using an output function f_o.

way so that a defined input pattern transforms to a corresponding output pattern when routing the neuron signals from the input to the output layer. It therefore applies an input pattern from training to the input layer and transforms the information through the hidden layers to the output layer (forward propagation). Then, it measures the error between the actual and the expected output pattern from training data for each neuron in the output layer. This error is propagated backwards towards the input layer and adapts the connection weights according to the the propagated, weighted error (backpropagation). Each weight is adapted according to the error, so that the total error is smaller for this sample – with the amount of change usually being controlled by a learning rate $\alpha > 0$.

4.4 Increasing Classification Accuracy

We currently do not utilize boosting or bagging of classifiers within the implementations conducted for this thesis, but might likely incorporate corresponding functionality in future research as it will likely increase face detection and/or recognition accuracy.

4.4.1 Boosting of Classifiers

The concept of classifier boosting is to combine multiple weak classifiers to a single, strong classifier incorporating a modified learning procedure. A

[2] Feed forward neural network fetched from http://commons.wikimedia.org/w/index.php?title=File:Artificial_neural_network.svg&oldid=89256918 on 2013/08/18.

weak classifier is a classifier not covering all data, but focusing on classifying parts of the data correctly. Therefore, it eventually classifies major parts of samples wrong, but classifies some samples correct. In order for boosting to work, contained weak classifiers just have to classify with higher accuracy than a random classifier (therefore incorporate knowledge about the data). Strong classifiers are classifiers composed out of multiple weak classifiers that cover all parts of the data. They therefore stick closely to the correct classification results with major parts of the test samples. In general, boosting approaches tend to use a part of the training data to train a weak classifier C_1, then give increased weight to/reuse samples classified wrong by this classifier when training the next classifiers C_N. This way wrongly classified samples probably are classified correctly by C_N. Further, classification results of weak classifiers are weighted and combined to obtain a single, strong classifier result, e. g. using majority voting. Consequently, the initial wrong classification of a sample by C_1 can conceptually be outweigh by the correct classification from C_N. Consequently, boosting can just improve classification accuracy if the weak classifiers actually only cover a part of the dataset accurately.

On the one hand, boosting proved useful in many evaluations of complex pattern recognition problems. On the other hand, the boosting comes with increased effort in therms of learning complexity and possibly duration. In comparison to classifier bagging – which can combine independently and ready trained classifiers (they can even utilize totally different, independent data) – boosting is conceptually performed on the same training data set and eventually involves an adapted training procedure.

Important research in boosting was conducted by Freund and Schapire with AdaBoost [68, 70], which was the conceptual basis for further development, including Real AdaBoost by Schapire and Singer [159], LogitBoost and GentleBoost by Friedman et. al. [71–73], CoBoosting by Collins and Singer [46], BrownBoost by Freund [67] or RankBoost by Freund et. al. [69]. Further, Rosset [150] describes robust boosting using a weight decay and states the relation of his approach to bagging in detail. A projection of most difficult examples instead of using a random subspace is used by García-Pedrajas et. al. [79] to create consecutive training datasets in order to increase accuracy boosting accuracy. Li [117] proposes abc-boost, which adaptively and greedily chooses a base class for each boosting iteration. Based on multiple additive regression trees (MART) [71, 72] they implement and evaluate abc-boost as abc-mart. Li et. al. [121] boost SVM based classifiers using AdaBoost. They further discuss why boosting conceptually strong classifiers is useful as well as boosting simple classifiers.

4.4.2 Bagging of Classifiers

With classifier bagging multiple classifiers are combined to a single, more comprehensive classifier using model averaging (e. g. Breiman [28, 29], Quinlan [146], Skurichina and Duin [172, 173], Ditterich [57] and Kuncheva [111]). Bagging is used to a) obtain scalar classification results from multiple classifiers, possibly classifying very different data, and b) to improve the classification stability and accuracy. There is one condition classifiers intended for combination by model averaging must fulfill in order for bagging to increase results: the classification must be unstable, so that changing the excerpt of data used as train data significantly changes the classification results. In comparison to boosting, which is designed to improve classification rates by incorporating a modified learning procedure on the same dataset, bagging my combine ready trained classifiers eventually even trained and classifying very different types of features and datasets.

Chapter 5

Our Approach

In this chapter we present a stereo vision based pan shot face unlock for mobile devices. Based on this conceptual pan shot face unlock idea, we describe and evaluate different stages of our approach in detail in chapter 7.

Our aim is to extend mobile device authentication by combining all sensor information that is available from a pan shot of the mobile device around the user's head (moving the mobile device 180° from left profile over frontal to right profile of his or her face, see figure 5.1) – in particular the (2D or stereo-vision-3D) device camera and the movement sensor data from accelerometers, gyroscopes, and magnetometers. This approach is still fast and convenient to use but harder to circumvent by a photo attack than with using frontal perspective face information only – as more information than contained in a frontal picture of the face would be needed (i.e., attackers would need to provide a 3D reconstruction of the person's face or a closely synchronized video stream instead of a single, static photograph).

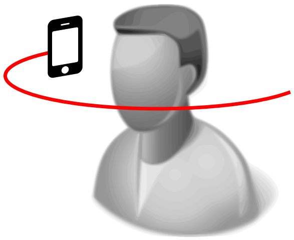


Figure 5.1: The mobile device records the user's face during a pan shot.

5.1 Intended Pan Shot Face Unlock Usage

A pan shot face unlock requires a mobile device with a frontal camera and sensors such as a gyroscope. Our aim in terms of usability is a quick swipe of the user's mobile phone around the front side of their head: the user holds the mobile phone either right or left of his or her head, so that the frontal camera points towards one ear. The arm holding the phone should be stretched. Then the user moves the mobile phone in a rough circle via the frontal view along to the other side of the head, so that the frontal camera points towards the other ear. The arm holding the phone should be kept stretched. All data obtained by the mobile phone, including data recorded by the frontal camera and motion sensor time series, is then used for face authentication.

5.2 Pan Shot Face Unlock Toolchain

Our proposed pan shot face unlock toolchain (see figure 5.2) at first records pan shot data by using a stereo camera and a gyroscope sensor integrated into the device. Using a stereo to range algorithm, range images are composed, each out of a pair of stereo images. We then perform error correction on obtained range images to repair erroneous regions. Further, we use sliding window based template matching as approach to face detection. Therefore we first create templates of the average torso and head from different perspectives, then search for best matching image regions using the templates of the corresponding perspective. In order to precisely segment face related from background information, we at next cut out the face using either the corresponding face template contour, or GVF snakes fitted to the individual's face contour. Based on the (now segmented) pan shot faces from different perspectives, we apply face recognition using multiple classifiers (a classifier per perspective). We first train a face classifier using segmented grayscale and range faces from the corresponding perspective, then perform face recognition to obtain a classification result. In order to combine face classification results from different perspectives, we finally perform majority voting as a form of classifier bagging.

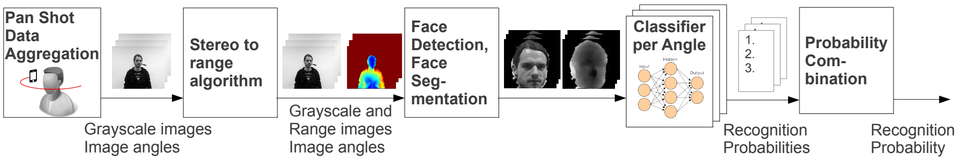


Figure 5.2: Overview of modules used in the stereo vision pan shot face unlock toolchain.

5.3 Pan Shot Data Aggregation

To record data from a 180° pan shot of the device around the user's head we utilize built-in cameras and sensors. In particular, when users hold the device with the camera facing their left or right ear for starting the unlocking process, we intend the device to start recording data. Conceptually, the device can start this recording automatically, which requires it to recognize the user's intention at this point. (e. g. using ear recognition). Further, the device needs to inform the user about recording a pan shot now being started. We have not investigated in automatically starting a pan shot data recording yet (will be subject to future work), but require the user to start the pan shot, e. g. by pressing a button on the mobile device. During the pan shot, the device continuously records data. For visual data this can be done in the form of recording a video stream, or – as for sensor data – periodically, by recording images. When making use of sensor data, especially gyroscope sensors, while the recording is still in progress, it is also possible to record images and sensor data after the device has rotated a certain angle α. This way there will be a record each for multiple perspectives around the user's head: e. g. recording new data with $\alpha = 15°$ will produce 12 records for a 180° pan shot. We use $\alpha = 30°$ for our first implementation, then change the angle to roughly fit the perspectives data has been recorded within the u'smile face database by using $\alpha = 22.5°$. Before further processing recorded data, the angles of records are normalized, so that the angle of the frontal perspective is roughly set to 0°, the angle of records covering the left side of the user's head being negative and the angle of records covering the right side being positive. As the user finishes the pan shot, the device is naturally facing the user's other ear. At this point, the device can conceptually stop the recording automatically – but as for starting the pan shot recording within our approaches the user is required to press a button on the device to stop recording.

5.4 Range Image Creation

Using the grayscale stereo vision image pairs from multiple perspectives obtained during a pan shot, we create a range image from each pair of stereo vision images. The techniques used within these stereo vision algorithms are beyond the scope of this thesis, as they are investigated in parallel in [190]. Each range image contains information related to the camera-object distance in each pixel. Therefore, these range images show the user performing a pan shot face unlock in the range domain (see figure 5.3).

Range images derived from a pair of stereo vision images likely contain erroneous regions, such as areas not containing range information and areas containing incorrect range information. These errors are often caused by

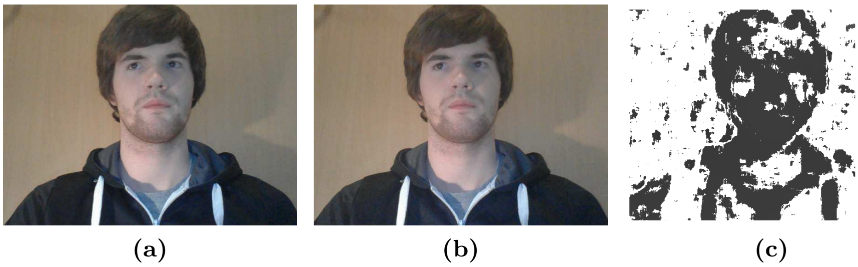


Figure 5.3: Frontal perspective of a user performing a stereo vision based pan shot face unlock with a) and b) showing the left and right camera images and c) the derived range image [63].

the concepts used in the stereo to range algorithms. E. g. areas not visible in both images and regions too homogeneous to correlate pixels within the two images will result in areas not containing range information. Caused by these erroneous regions, further errors will arise during processing these images within face detection and recognition. In order to avoid these errors we perform a range error correction on obtained range images before further processing them:

- To correct areas containing incorrect range information (such as peaks or holes), low pass filters can be used.
- To correct areas not containing range information hole filling algorithms can be used. These often derive pixel information from the pixel neighborhood (surrounding pixels), such as using interpolation or a k-nearest-neighbor algorithm.

Using these error corrected range and the original grayscale images, we perform face detection and segmentation as the next step.

5.5 Range Face Detection and Segmentation

Based on recorded visual and sensor data, we perform face detection and segmentation in order to obtain visual data as purely related to facial information as possible, which we then use for face recognition. As recorded visual data shows the user's head and face from multiple perspectives, we need to take into account the perspective of each image/video frame when processing it. We utilize different approaches to face detection and segmentation. At first, we use Viola and Jones face detection based on Haar-like features [115, 186] and rectangular cropping for face segmentation. To address finding faces from multiple perspectives, we make use of different feature cascades of the Viola and Jones object detection framework, namely for detection frontal and profile faces (see figure 5.4).

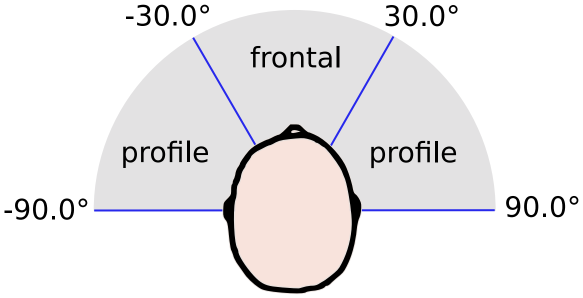
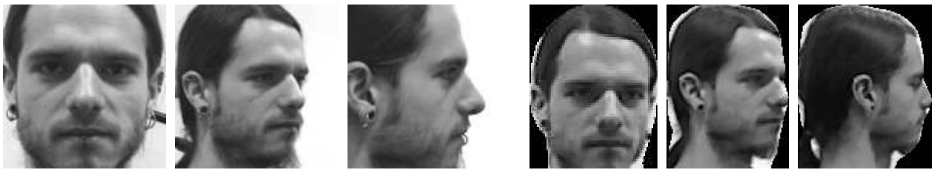


Figure 5.4: For detecting faces from multiple perspectives using Viola and Jones face detection [115, 186], the chosen feature cascade depends on the angle of image recording.

Although face detection results based on the Viola and Jones approach might seem adequate for successive face recognition, they incorporate several problematic factors – including a) unequal normalization in terms of face size and position and b) background information included in face images due to rectangular cropping of faces. We therefore use range based template matching as second approach to face detection and a template based cutout along with GVF snakes [196] for precise face segmentation. We therefore first create range templates showing the user's head from multiple perspectives. For performing template based face detection, we search for the best template match in a range image using sliding window template matching. Then we cut out the face either along the template contour or use GVF snakes to precisely find and segment along the actual face contour. Using segmented face images (see figure 5.5) along with sensor data, we perform face recognition as the next step in the face unlock toolchain.



(a) Viola and Jones based

(b) Template matching based, discarded areas marked black

Figure 5.5: Face images segmented using different detection and segmentation approaches as used for subsequent face recognition.

5.6 Face Recognition

Based on detected and segmented grayscale and range face images from multiple perspectives, we now perform face recognition. We therefore use separated view-based classifiers for each perspective – as well as for grayscale and range data. Assuming 9 perspectives equally distributed in a 180° pan shot with a grayscale and range image each, we use 18 separate face classifiers in total with each covering an angle of 22.5° (see figure 5.6) – but using a different amount of classifiers is possible as well. The classifier an image corresponds to is chosen by the image type and normalized pan shot angle of recording.

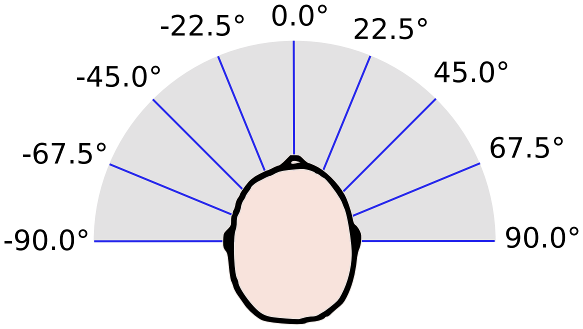


Figure 5.6: Perspectives at classifiers are distributed in a 180° pan shot when using 9 classifiers per image type. The classifier perspectives are stated relative to the 0° frontal perspective.

We utilize the standard approaches of support vector machines (SVM) and neural networks (NN) as face recognition classifiers. Using labeled data (face images for which the people's identity is known) we train the classifiers in order to distinguish later between the identities of unlabeled face images. Our pan shot face unlock approach requires the classifiers to distinguish between two types of users: those allowed to interact with the mobile device (legitimate users, typically one), and those not allowed to interact (illegitimate users, typically many). Therefore, we treat the face classification as a binary classification problem: faces of the legitimate user form the positive class, and faces of illegitimate users form the negative class. Consequently, after a user performs a pan shot face unlock, the classifiers have to decide if the user is a) a legitimate user and will be allowed to further interact with the device or b) an illegitimate user and will not gain device access – both without deriving the actual identity of the person from the aggregated data. To chose the classifier fitting this problem best, we evaluate differently configured classifiers using training and test data throughout our implementations (see chapter 7).

In order to successfully classify a yet unknown illegitimate user correctly, our approach conceptually requires to include a large amount of diverse face images to the negative class. As we don't want to bother users of our unlocking approach to provide this data by themselves, we intend to include diverse sets of negative samples within our implementation. This way users only need to provide samples for the positive class themselves, which will most likely be images of their own.

5.7 Combining Classifier Results

From classification results of grayscale and range face classifiers from multiple perspectives we derive a single face unlock result in the next step. Conceptually, there exist multiple approaches to combining these results, such as forming the weighed sum or weighted product of classification probabilities. In our approach, we rely on a simple but effective majority vote. Each classifier votes for the user either being a legitimate or illegitimate user, with the choice selected through the majority of votes being the final results. We distinguish between a weighted and non-weighted majority vote at this point:

- With a non-weighted majority vote the choices of each classifiers weight the same. Therefore choices of classifiers which proved to successfully distinguish between legitimate and illegitimate users during the test phase will matter the same as from classifiers which performed worse during testing.
- With a weighted majority vote the choices of classifiers will be weighted e. g. by their classification performance during testing. Consequently, choices of classifiers which proved to successfully distinguish between legitimate and illegitimate users during the test phase will matter more than from classifiers which performed worse during testing.

On the one hand, using a weighted majority vote may improve the classification accuracy, as classification results with a low probability of being incorrect will weight more than of those with a higher probability of being incorrect. On the other hand, using a weighted majority vote will lead to a less homogeneous distribution of classification result weighting, which will further result in some classifiers being more important for actually unlocking the device than others. In addition, this could lead to attackers preferably targeting perspectives which are expected to be more important for unlocking the device. In the hypothetic case of the major responsibility for unlocking the device being held by a single classifier, namely the grayscale, frontal perspective classifier, our stereo vision based pan shot face unlock will basically be reduced to a simple, frontal grayscale face unlock – which fundamentally defeats the purpose of a pan shot face unlock. As we have not investigated this issue in detail yet, it will be in the focus of our future research.

Chapter 6

Test Data

To reproducible evaluate our pan shot face unlock approach, we create the u'smile face database. There are several reasons why we created this face database for our face recognition experiments with a controlled set-up instead of in-the-field with mobile phones: a) the illumination of faces shown in pictures taken with a pan shot around the users head varies strongly with each pan shot. Therefore, the test results would not be reproducible and comprehensive enough; b) in our experiments, the frontal camera photo quality strongly depended on how fast the user moved the mobile phone. Moving the mobile phone from one ear, along the frontal face perspective to the other ear took about 4 seconds to obtain photos of good quality. In case the user moved the mobile phone faster, the image quality was lowered due to motion blur, which consistently lowered the system reliability; and c) we are not aware of any other face databases available for research that contain face pan shots, state the angle at which a picture was taken, and have multiple pictures per angle and person available at the same time.

The u'smile face database was created in two stages: a first stage, containing grayscale images and designed to be a preliminary face database for aggregating first pan shot face unlock results and evaluating the face database usability for research. The second stage is based on lessons learned from the preliminary version and contains range data alongside grayscale data.

6.1 Preliminary Pan Shot Face Database

For doing first, comprehensive tests on our face recognition approach, we have created a preliminary face database at FH Hagenberg[1]. This database features 38 people with 1-3 pan shot image sets each and 95 such sets in total.

[1] We gratefully acknowledge the help of Christina Aigner, who performed the actual image recording in the context of her Bachelor thesis.

Each set contains 4-5 grayscale images[2], recorded at the angles 90°, 45°, 0°, -45° and -90°, with 0° being the frontal face perspective (see figure 6.1).

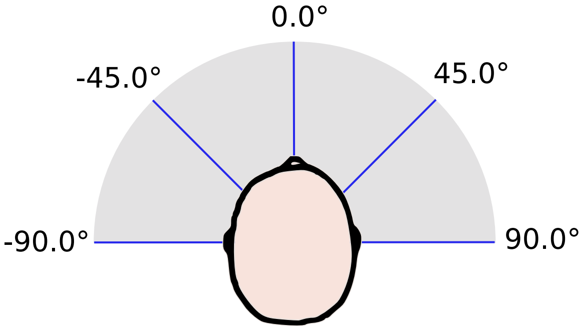
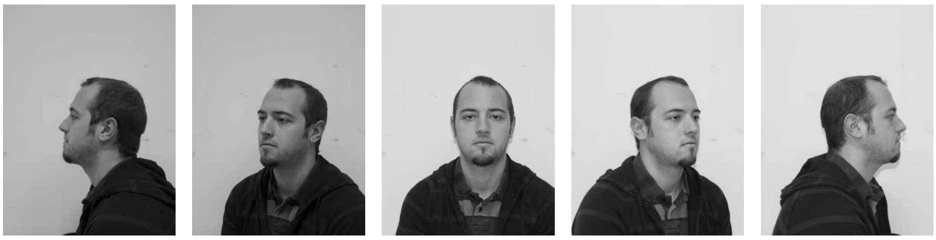


Figure 6.1: Perspectives at which images were recorded for the preliminary pan shot face database.

The images were recorded using a Nikon D50 camera. The unedited, original image dimension was 5184px × 3456px – during preprocessing, the size has been decreased to 281px × 375px due to resizing and cropping the image. The facial expressions of all images in a set are either normal, eyes closed or smiling and the illumination of the faces is evenly good. As participants were allowed to change their appearance themselves between pan shot sets (e. g. facial expression, usage of glasses, different style of hair, different clothing and jewelry) there is no definite correlation of pan shot set number and style. One such set is shown in figure 6.2.



(a) -90° **(b)** -45° **(c)** 0° **(d)** 45° **(e)** 90°

Figure 6.2: Pan shot image set from our preliminary face database.

Using our preliminary face database was made difficult for several reasons, which we want to present as a short roundup of lessons learned. First, our data featured only a low grade of normalization. This lead to complicated data preprocessing. Secondly, several data sets were incomplete due to

[2] For some pan shot sets, the image at 90° is missing.

no-standardized recording. Parts of these data were usable in the end, but other parts failed to be preprocessed or were insufficient for usage in classification and were excluded from the database. Finally, training and testing classifiers was difficult because of less data being recorded per subject.

Therefore, the lessons learned are: the recording setup needs to be documented. Recording condition documentation should include: time and location of recording (illumination), setup of the room (e. g. usage of white screens/linen), position of the participant (especially the position and rotation of the participant's head), the camera position and rotation as well as the camera zoom. Applicable systematic changes during recording should be documented too. This may include patterns of a) style changes (glasses, clothing, jewelry, hair), b) facial expression changes, c) changes in looking directions and illumination. Using normalized recording conditions (e. g. same camera and head position and rotation for all participants, same pattern of style changes) ease data preprocessing. Recording multiple complete data sets per subject is mandatory for evaluating our approach, as distinct treatment of perspectives requires classifiers to be trained and tested on data exclusively recorded from this perspective – which may further be originated by a single subject.

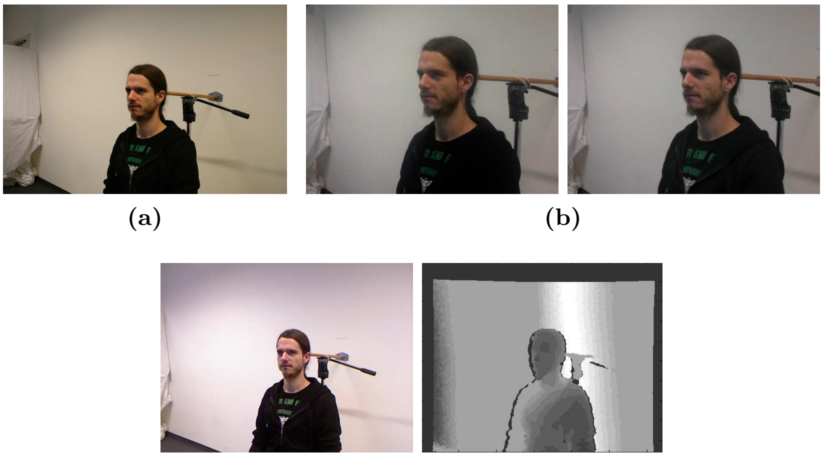
6.2 u'smile face database

Based on lessons learned from the preliminary face database and for doing comprehensive tests on our face recognition approach, we have created the u'smile face database. This database is designed to contain a wide variety of face data, starting with a data set provide test data for pan shot face detection and recognition with grayscale and range images (range images represent the camera to object distance as pixel values), with realistic indoor lighting conditions[3]. It contains 30 different people, each with 20 numbered pan shots and recorded with different devices. For each device, each pan shot image set features 9 different perspectives from one 180° pan shot around the user's head – each 22.5° an image/image pair has been taken, with 0° being the frontal face perspective (in correlation to the perspectives classifiers are arranged for face recognition, see figure 5.6). For each person, pan shot image set, and perspective, the following images are contained in the database (see figure 6.3):

- A high quality, colored 2D image, recorded with a digital single-lens reflex camera (Canon EOS 400D, 3888px × 2592px).
- A colored, 2D image pair, recorded with a mobile device stereo camera (LG Optimus 3D Max P720, 2× 640px × 480px).

[3] We gratefully acknowledge the help of Christopher Gabler, who performed the actual image recording in the context of his Bachelor thesis.

- A colored 2D and a raw range image, recorded with a Microsoft Kinect and the OpenKinect framework (2× 640px × 480px).



(a) **(b)**

(c)

Figure 6.3: Excerpt of the u'smile face database showing one dataset for one subject from one perspective, with a) high quality, colored 2D images, b) colored 2D mobile device stereo camera image pairs and c) OpenKinect colored 2D and range images.

For each pan shot image set, the direction and facial expression was slightly varied by the participant to give some variety for the training data. Table 6.1 states the relation between pan shot image set number and the participants' direction/facial expression (from the participant's perspective), with an already preprocessed example shown in figure 6.4.

Various parts parts of the u'smile face database can be obtained for research and teaching with attribution to [63] from the u'smile project homepage (online at http://usmile.at/downloads).

6.2.1 Recording Setup Description

Recording took place indoors at the University of Applied Sciences Upper Austria, School of Informatics, Communication and Media in Hagenberg. Recording was done with realistic indoor illumination conditions: the main source of light was artificial light from above; additionally, sunlight was indirectly shining into the room from glass partition vis-a-vis of the participants. The recording devices where mounted on a wagon in about the height of the sitting person's head (see figure 6.5a). The wagon was rolled 180° around the sitting user to record a full pan shot, keeping a distance of about 1.5m to the head. The 9 wagon positions of recording were marked on the floor to obtain the exact same 9 perspectives for all pan shot sets (see figure 6.5b).

To have the head at the roughly same position for all participants, we used a wooden stick mounted on the wall, against which each participant lent his or her head (see figure 6.5c). This stick causes an artifact in the data, visible in both profile perspectives for all devices. The five positions users were supposed to systematically look at during recording were marked on the opposite wall (see figure 6.5d).

6.2.2 Lessons learned

Compared to the preliminary face database, this database version contains many more (20) pan shot data sets per participant. Still, comprehensive training and tests are limited by the amount of data per subject – therefore recording even more data per subject would be required and will eventually be done in the future. As the recording setup was well normalized and documented, the variance in data is low. E. g. participants did not significantly change their head positions between pan shots. On the one hand, such a tight grade of normalization eases processing the data and normally increases later classification accuracy. On the other hand, classifiers possibly learn features based on side effects of this high grade of normalization additional/instead of learning subject related features. As example, classifiers may learn the exact position of a subject's head, which varies little within the subject's data, but varies more within different subjects (due to initial positioning variances). Consequently, classifiers may distinguish subjects by using the exact position of the head only – without actually making use of subject related features. The same applies to illumination conditions: due to all recordings of a participant being done at once and the sunlight added to the artificial light slightly changing over time, the illumination conditions changed continuously throughout the subjects' recordings – with all data of a single subject still roughly showing about the same illumination. Classifiers could make exclusive use of the illumination to distinguish subjects. Therefore a certain amount of irregularities in data normalization may be useful for future recordings to suppress these types correlations. These irregularities might be added after recording in artificial way, e. g. by downscaling/blurring, adding a certain amount of noise, random position shift or scaling.

Nr.	Look direction	Facial expression
0	straight	normal
1	straight	smiling
2	straight	eyes closed
3	straight	mouth slightly opened
4	slightly top left	normal
5	slightly top left	smiling
6	slightly top left	eyes closed
7	slightly top left	mouth slightly opened
8	slightly top right	normal
9	slightly top right	smiling
10	slightly top right	eyes closed
11	slightly top right	mouth slightly opened
12	slightly bottom right	normal
13	slightly bottom right	smiling
14	slightly bottom right	eyes closed
15	slightly bottom right	mouth slightly opened
16	slightly bottom left	normal
17	slightly bottom left	smiling
18	slightly bottom left	eyes closed
19	slightly bottom left	mouth slightly opened

Table 6.1: Look directions and facial expressions for each pan shot.

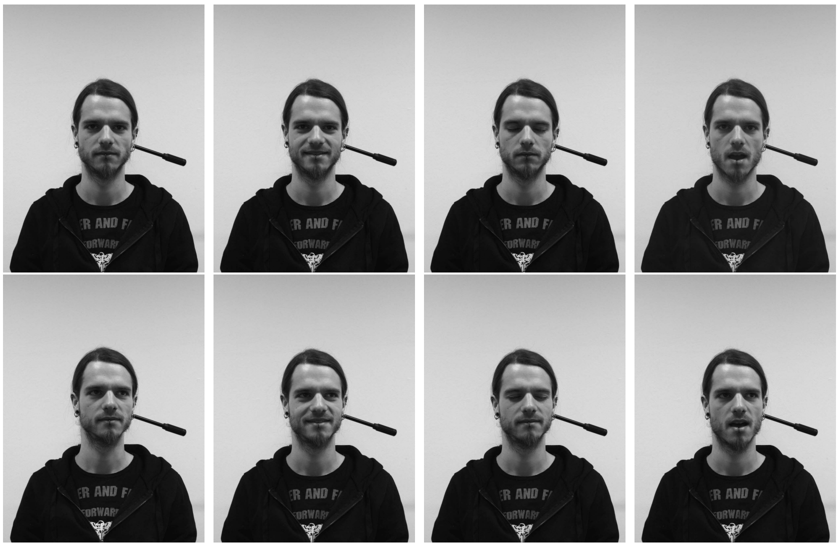
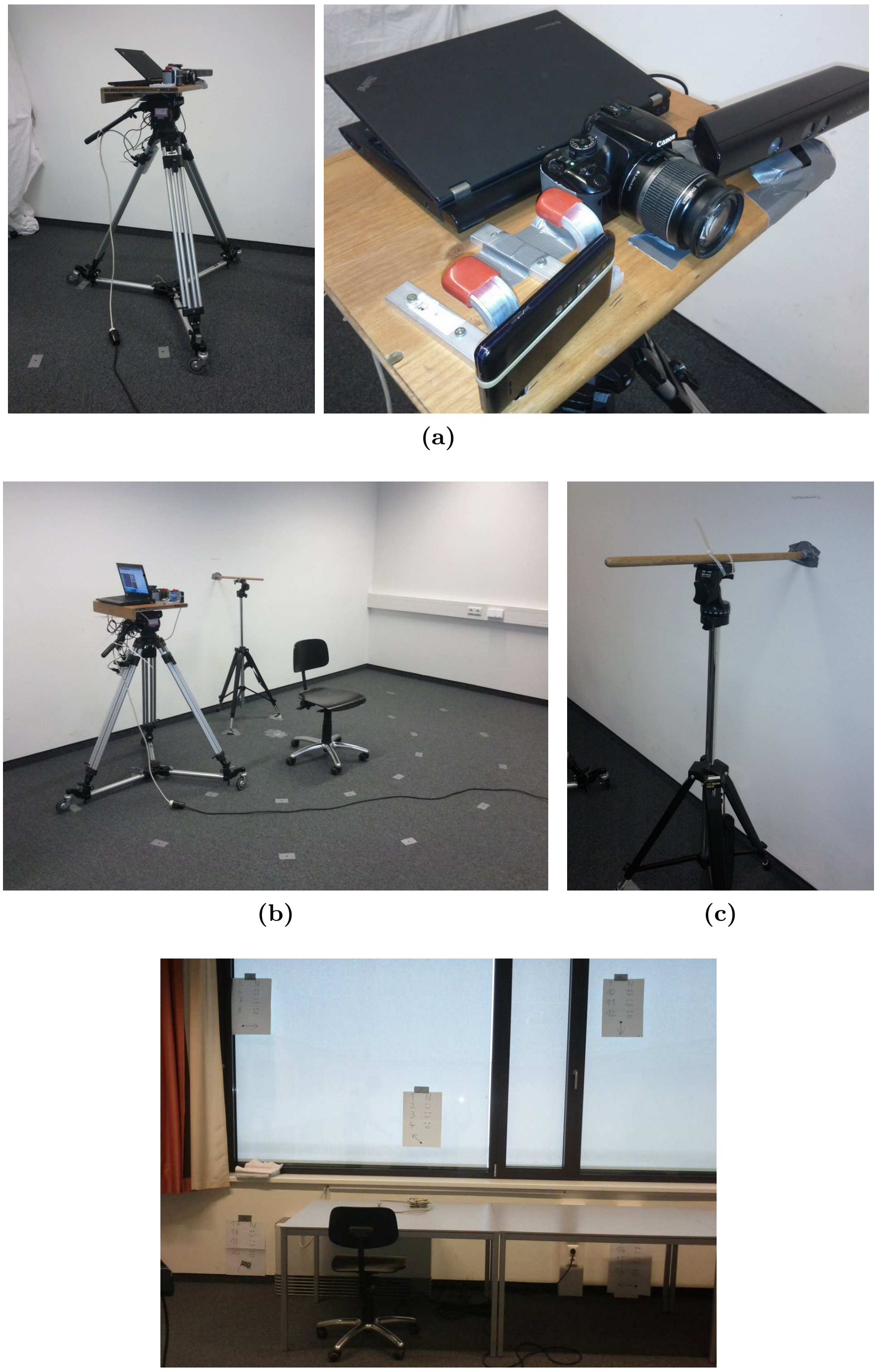


Figure 6.4: Preprocessed images of 8 different pan shots (Nr. 0-7), featuring different look directions and facial expressions.



(a)

(b)

(c)

(d)

Figure 6.5: u'smile face data base recording with a) recording devices being mounted on a movable wagon, b) the recording positions marked on the floor using tape, c) the stick used for positioning participants' heads and d) the look directions being marked on the opposite wall.

Chapter 7

Implementations and Results

In this chapter we describe the prototypes implemented along developing our mobile device pan shot face unlocking approach. The implementations were done in stages, so that weaknesses of each stage were identified and improved within the next stage. The first state (see section 7.1) introduces a prototypical pan shot face recognition on Android, using Haar-like feature based Viola and Jones face detection, Eigenfaces for face recognition and a our preliminary face database for conducting an evaluation. With the first stage we identify the face recognition and small amount of data as main components deserving improvement. Consequently, in stage two (see section 7.2) we evaluate support vector machines (SVM) and neural networks (NN) as face classifiers, introduce the u'smile face database and use it for more comprehensive tests of our face unlocking approach. Within stage two we identify the face detection as not working reliably enough for detecting a single, upright face in a mobile face unlocking scenario. In stage three (see section 7.3) we therefore extend our face unlocking approach to make use of range data (range images composed from mobile device stereo camera images). We introduce a range template based face detection and segmentation and evaluate our new approach on range images of the u'smile face database. In stage four (see section 7.4) we improve the range template creation, template matching and face segmentation. We further extend the approach with using GVF snakes for precisely segment the user's face along its actual contour and again evaluate our approach using range images contained in the u'smile face database.

7.1 Android Prototype (Proof of Concept)

In this section we present a first prototypical implementation towards a mobile device pan shot face unlock using an Android smart phone, which was first presented in [65]. The intention of this prototype is to use recorded real life data and use it for an proof-of-concept on-device user identification.

For aggregating grayscale pan shot data we utilize the frontal camera and gyroscope sensor integrated into the mobile device. In order to extract faces from recorded images, based on the image angle we either perform frontal or profile face detection using the approach of Haar-like features by Viola and Jones [115, 186]. Based on the image angle we then perform face recognition using Eigenfaces by Turk and Pentland [181]. Finally, we sum up the probabilities from classifiers of different perspectives in order to obtain a scalar recognition probability per subject.

7.1.1 Method

Intended Usage

The Face Unlock we intend to develop requires a mobile device with a frontal camera and sensors such as a gyroscope. Although our mid-term aim in terms of usability is a pseudo-3D reconstruction of facial features with a quick, non-standardized swipe of the user's mobile phone around the front side of her/his head, in this first stage of prototypical implementation we require the user to perform a more formalized swipe of the camera: the user holds her/his mobile phone either right or left of her/his head, so that the frontal camera points towards the ear. The arm holding the phone should be stretched. The user then moves the mobile phone in a rough circle via the frontal view along to the other side of her/his head, so that the frontal camera points towards the other ear. The arm holding the phone should be kept stretched. The data obtained by the mobile phone, including a frontal camera video stream and motion sensor time series, is then used for Face Unlock to avoid the simple attack vector of presenting a static picture of the user's face to a static phone.

Pan Shot Face Recognition Toolchain

During a pan shot, different images of the user's head are recorded from different perspectives. Using these images and the angle they were taken at, we performed either frontal or profile face detection – which results in the extracted faces along with the angle at which were originally recorded. We use this data to a) train classifiers and b) classify new face images. For different angles we use different classifiers, so that each classifier only covers a certain angle during training and classification and can therefore specialize for this point of view. For each subject, the classification is treated as a binary classification problem: the subject's face images are the positive class, the face images of all other subjects are the negative class. For each pan shot, the classification results for different angles are combined to a single scalar value – estimating the overall probability of having detected an authenticated or a non-authenticated user. Figure 7.1 provides an overview of this toolchain, as it has been implemented for the Android prototype in this stage and is simulated for improvements on desktop computers in the succeeding stage.

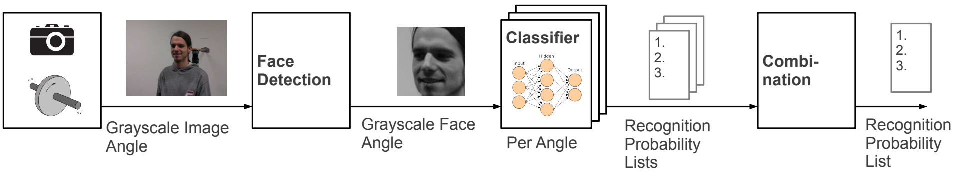


Figure 7.1: Overview of the modules used in the pan shot face unlock toolchain.

Environment

As environment for a Face Unlock prototype we are targeting an state of the art mobile phone with frontal camera and at least a gyroscope and based on Android to enable future integration into the platform unlock feature. For the implementation within this first stage, we use a Google/Samsung Nexus S GT-I9023 device running Android 2.3.3. For face detection and recognition we use OpenCV [25] compiled for Android[1] and JavaCV for Android[2] as wrapper around OpenCV.

Face Detection

The Face Unlock application has a state `STATE`, which initially is `IDLE`. As the user holds the mobile phone with the frontal camera towards one ear, the application changes from `IDLE` to `ACTIVE`. The application stays active as the user moves the mobile phone via his or her frontal face towards the other ear. As soon as the frontal camera points to the other ear – determined by the gyroscope data –, the application goes from `ACTIVE` to `IDLE` again. In our prototype implementation, changing `STATE` is done by the user pressing a button. As long as the application is `ACTIVE`, photos are taken using the frontal camera. The application decides when the next photo should be taken by monitoring the device angle, resulting from the gyroscope time series. If the changes in the device angle since the last photo are larger than a defined threshold α, the next photo is taken. For our experiments, $\alpha = 15°$ has been used. Each photo is stored along with metadata (most importantly the current device angle). Therefore, roughly the same number of photos are made for a pan shot done for each Face Unlock, and processing the photos can be done afterwards.

[1] http://opencv.alekcac.webfactional.com/downloads.html
[2] http://code.google.com/p/javacv/downloads/list

We do not record a full video stream of the whole camera movement across the user's face because of the mentioned limitations in the mobile phone APIs: on the one hand, most phones offer only limited resolution in video mode when compared to picture mode, and on the other hand, Android does not yet support accessing the raw video stream with low processing overhead from third-party applications. Additionally, the limited processing resources on current mobile phones would not allow to process the full video stream for face recognition in real time.

As the application switches from `ACTIVE` to `IDLE`, the following steps are processed: first, a normalization of the metadata stored with each photo is performed. Assuming that – seen from a frontal face perspective – the user has held the mobile phone at roughly the same angle when starting and ending the Face Unlock, the frontal face perspective is defined to be at an angle of 0°. When β is the total angle the mobile phone has rotated, the normalization is performed so that the maximum left angle of all photos is roughly $-\frac{\beta}{2}$, and the maximum right angle of all photos is roughly $\frac{\beta}{2}$. Second, all photos are converted to gray scale. This conversion incurs some information loss, but most face recognition algorithms operate on gray scale only to be more robust against different lighting conditions, and the limitation to a single channel allows faster processing in subsequent stages.

Finally, face detection is performed for each photo. The OpenCV face detection classifier cascades is chosen depending on the metadata stored along with each photo, where γ is the device angle the photo was shot at and ϕ is a predefined threshold angle. If $\gamma < -\phi$, the `PROFILE` classifier cascade is chosen. If $\gamma > \phi$, the picture is mirrored[3] and the `PROFILE` classifier cascade is chosen. If $|\gamma| \leq \phi$, the `FRONTAL-ALT` classifier cascade is chosen. For our experiments $\gamma = 30°$ was used. Face detection is then performed using the chosen classifier. Finally, areas that are found to contain a face are extracted from the pictures and saved to separate face images along with the angle the picture has been taken at. Figure 7.2 shows the pictures recorded during one pan shot, along with the faces detected in those pictures. These face images are then used for face recognition in the next step.

Face Recognition

For face recognition, the Face Unlock application contains several classifiers. Each classifier covers a certain angle-of-view α of the user's face, which corresponds to the multi-view approach of Pentland et. al. [142]. Therefore, face images shot at a similar angles will be assigned the same or a neighboring classifier in the normal case. For our experiments, we used $\alpha = 20°$, which results in about 9 classifiers for an assumed total device rotation of 180°.

The Face Unlock application can either be in `TRAIN` or in `CLASSIFY` mode. For both modes, the application takes the face images resulting from sec-

[3] The OpenCV `PROFILE` classifier cascade only detects left profile faces.

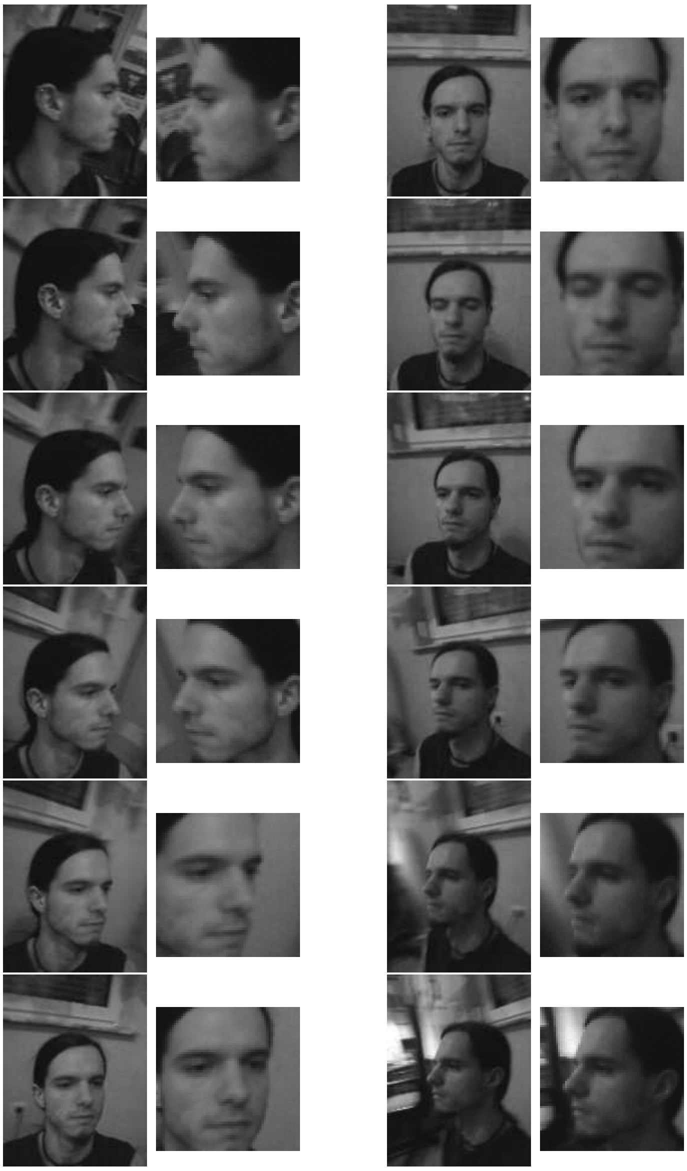


Figure 7.2: Pictures recorded and faces detected from one pan shot.

tion 7.1.1. In TRAIN, the classifiers are trained with face images of people that should later be recognized. Therefore the identity of the person is set manually in this mode. Detected face images are saved and assigned to a classifier, corresponding to their angle. As face classifiers we use Eigenfaces for face recognition as proposed by Turk and Pentland [181] in the first stage. Eigenfaces are based on an average face, which has to be recalculated every time the training faces are expanded — otherwise, no expansion will be possible.

As the user switches the Face Unlock application from TRAIN to CLASSIFY, each classifier is trained with all face images assigned to it. This training is done on the end-user phone without requiring server-assisted (“cloud”) computation for privacy reasons. Our prototype implementation is therefore also a proof of concept of the feasibility of on-device biometric authentication on current smart phones.

When the Face Unlock application is in CLASSIFY mode, detected face images are classified by the classifier corresponding to the angle at which the face image has been shot. For each face image to classify, a classifier delivers a list of distances. Each distance corresponds to the difference of the face image to classify to the face images of the people known to the classifier. Probabilities of how certain the person currently unlocking the device is a person known to the system can be derived from these distance lists.

For our proof-of-concept Android implementation, we are summing up the probabilities of different angles to obtain an overall probability, with which access to the mobile phone can then either be granted or denied.

7.1.2 Test Setup and Results

Using the images from the our preliminary face database as input to our Face Recognition system results in a face detection rate (true positives in terms of authentication systems) of 100% for frontal face images (which use the FRONTAL-ALT classifier) and 90.5% for face images shot at angle γ and $|\gamma| = 45°$ and $|\gamma| = 90°$ (which use the PROFILE classifier). A few false positive cases from the PROFILE face detection, such as the examples shown in figure 7.3, negatively influence the latter face recognition, as they are used for training and test data as if they were correct results.

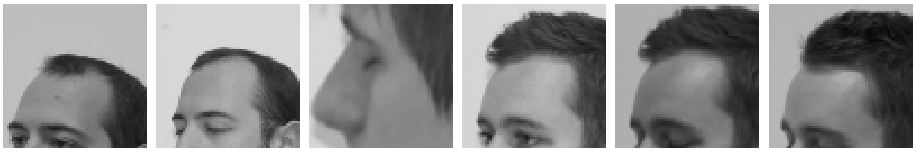


Figure 7.3: Examples for false positive results from the PROFILE face detection.

In the regular case a face can be detected in each picture taken in the pan shot, assuming a slow enough device movement of about 4 seconds for the total pan shot, and a sufficient illumination of the face, as shown in figure 7.2. In case of poorer illumination of the face, for some recorded pictures no faces might be detected, as shown in figure 7.4.

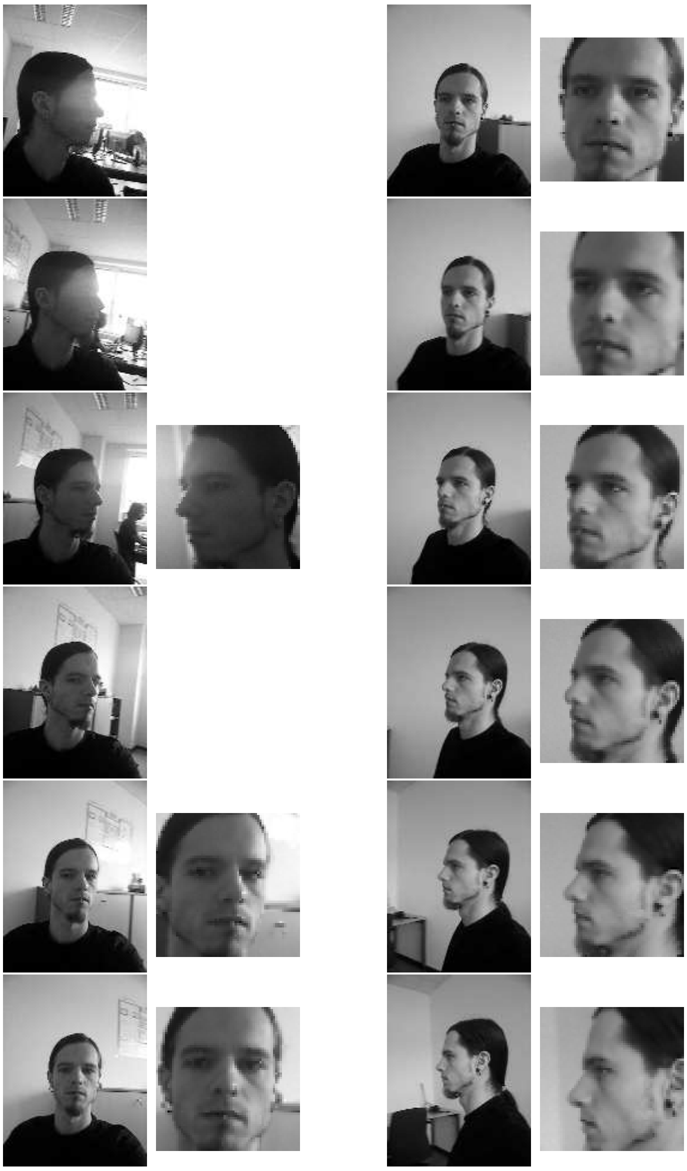


Figure 7.4: Pictures recorded and faces detected from one pan shot with more difficult illumination conditions.

Even if this detection rate is sufficient for our first, prototypical usage, improvements to the face detection might become necessary in the future, as more intensive tests of the algorithm in [58, 158] have shown that specially the profile face detection classifier of the OpenCV implementation suffers from a decreased detection rate.

The face recognition rates of our Face Unlock application have been evaluated using our preliminary face database in a test classification as follows:

1. Randomly chose a pan shot as test set. The person shown in this set is the test subject.
2. Chose other pan shots of the test subject as training sets.
3. Further add random pan shots of other persons to the training sets, until the training set contains 20 pan shots.
4. Train the Face Unlock application with the training set. The classification problem is reduced to a binary classification problem by treating all images that belong to the test subject as being part of the positive class, and all images of all other persons as being part of a single negative class.
5. Test the Face Unlock application with the test set.

For 100 such classifications, the test subject got recognized in 78.5% using frontal face information only, and in 55.8% using pan shot face information. We argue that the overall recognition rate (i.e. the true positives rate for the authentication case) is lower when using pan shot information because of our combining probabilities of the different views by simply summing up: as frontal face pictures seem to be easier separable than profile face pictures, as stated e.g. by Santana et. al. [158], and the profile faces are being detected less reliable at the same time, the better results of frontal face recognition are extended by a 4 times larger amount of worse results from profile face recognition. However, we assume a significantly higher difficulty level for tricking the system into authentication when relying on the pan shots instead of only frontal face shots. An attacker would have to replay a synchronized video stream while moving the attacked device or manufacture a 3D bust of the owner's face. Although we can not yet quantify the resulting increase in security, we argue that a small decrease in recognition rate is outweighed by the increase in security, which would support the day-to-day use of Face Unlock even for application scenarios with higher security demands.

7.1.3 Discussion

Our first results strongly indicate that face detection can be done sufficiently reliable even for pan shots, but that the second step of recognition based on Eigenfaces does not (yet) work reliably enough for further usage. Additionally, as stated by Belhumeur [13], changes in illumination are a major problem for recognition based on Eigenfaces. As changes in illumination are omnipresent in the mobile domain, this approach is not sufficient.

7.2 Improving the Android Prototype

In this section we present conceptual improvements to approach used with the proof-of-concept Android prototype (see section 7.1), which were first presented in [62]. In the first stage we have identified face recognition being the main component deserving improvement, especially as Eigenfaces for face recognition deliver inadequate results when used within conditions typical for usage in the mobile domain. Consequently, improvements include the usage of Neural Networks [132] and Support Vector Machines [143] as conceptually more powerful approaches to face recognition – as well as a significantly improved test set for evaluation. For face detection we still use the approach of Haar-like features by Viola and Jones [115, 186].

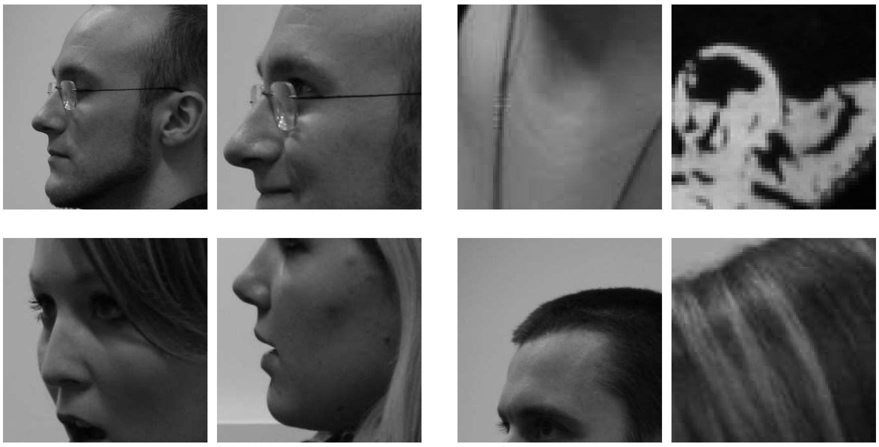
7.2.1 Test Setup and Results

To evaluate the benefit of improving specific parts of the toolchain used in the prototype, we simulate the toolchain with desktop computer scripts in Matlab and R. With these scripts we measure the performance of more promising approaches to face recognition, using 2013' Kinect color and range images from the u'smile face database as data source. Our face detection approach is evaluated on 620 instead of 600 pan shot sets, as we added 20 additional pan shot sets from a previously recorded person – with changed beard style. Our face recognition approach is evaluated using the standard 600 pan shot image sets.

Face Detection

Before performing face detection, we preprocess the images of our face database by cropping and scaling, then converting to grayscale. The preprocessed images are of 1000px × 1333px size, with large parts of the left and right side of the image being pruned. This is done in order to a) reduce the calculation power needed for processing the images and b) obtain an image layout and quality more realistic for images originating from a mobile device camera. We then perform face detection (including mirroring of right profile faces) as described in Recording Data and Performing Face Detection and cut out the biggest face found – if there is such one. In order to give a measurement of correctly/incorrectly or not detected faces, it is necessary to decide on a border between still correctly and just incorrectly detected faces (see figure 7.5). As deciding if a face should still be counted as detected correctly is a) hard to be done automated, and b) such a component will not be needed for a productive pan shot face unlock system, we did this decision by hand for all detected faces. The obtained face detection results for each perspective are stated in table 7.1.

The face detection results show clearly that, especially for non-frontal perspectives, many incorrectly detected faces will be passed to face classifiers in the next stage. This finding is consistent with more intensive tests of



(c) Correct **(d)** Incorrect

Figure 7.5: Deciding on a) still correctly detected faces and b) already incorrectly detected faces.

Persp.	Cor.	Incor.	Not.	Ratio
-90.0°	521	59	40	84.0%
-67.5°	567	47	6	91.5%
-45.0°	581	36	3	93.7%
-22.5°	374	132	114	60.3%
+00.0°	549	46	25	88.5%
+22.5°	398	132	90	64.2%
+45.0°	383	229	8	61.2%
+67.5°	532	49	39	85.6%
+90.0°	427	26	167	68.9%

Table 7.1: Face detection results for perspectives (Persp.): amount of faces detected correctly (Cor.), detected incorrectly (Incor.), nothing detected (Not.) and the correct to all ratio (Ratio).

the algorithm [58, 157, 158], which indicate that the profile face detection classifier of the OpenCV implementation suffers from a decreased detection rate. Consequently, this will lead to classifiers learning wrong data, as incorrectly detected faces are treated like faces as well during training and tests – and therefore adulterate overall face recognition results. Hence, we evaluate face recognition twice, with using a) correctly detected faces only, to evaluate the face recognition performance only, and b) using correctly and incorrectly detected faces, to obtain the overall system performance up to face recognition.

Face Recognition

Based on the detected faces and their angle of recording, we evaluate different face classifiers. In our preliminary experiments [65], we used Eigenfaces for recognition [181], with which we obtained an unsatisfying overall person recognition rate of 55.8% when applied to a preliminary face database of 38 people. Therefore, we now evaluate more promising approaches of differently configured support vector machines (SVM) and feed forward neural networks (FFNN) as face classifiers. Before performing face recognition, we resize found faces to 128px × 128px to have a uniform amount of features for each image, and reduce the amount of calculation power needed during processing. For adjusting the classifiers well, we do a parameter grid search for a) the number of hidden layer neurons for using FFNN classifiers, and b) configuring the SVM parameters corresponding to the used kernel, as suggested by Hsu et. al. [94]. We further treat our face recognition as a binary classification problem: for each of the 30 subjects from our face database, all images corresponding to the particularly selected subject represent the positive class – and the images of all other subjects represent the negative class. This results in the negative class being 29 times the size of the positive class – which consequently will lead the learning of our classifiers towards the negative class. As a result, the true positive rate will be lower than the true negative rate – which is according with our face recognition results. Each classifier is trained and tested on all of these 30 possible binary classification problems, with at most 60% of the data from the corresponding perspective, so that the other 40% are left for explicitly measuring the final classifier performance. Final results are measured in recognition rate distribution per classifier, and recognition rate distribution per angle for the best performing classifier.

Training and Test Procedure for Support Vector Machines For each angle, subject and classifier, one SVM is trained using the correspondent part of the train set, and evaluated on the correspondent part of the test set. For our implementation, we make use of LibSVM [39].

Training and Test Procedure for Neural Networks For each angle, subject and classifier, 10 FFNN are trained with the correspondent part of the train set. 30% of the total set of the corresponding perspective is used for training the network, 12% for cross validation to stop the training if improvements become too small, and 18% to evaluate the 10 generated neural networks against each other. Only the best performing network is evaluated on the last part of the test set afterwards.

Face Recognition Results Table 7.2 shows the configuration of the best performing classifiers and their corresponding parameter configuration. These classifiers were evaluated once each with a) using correctly detected faces only for training and test, and b) using incorrectly detected faces as well as correctly detected faces for training and test (see figure 7.6).

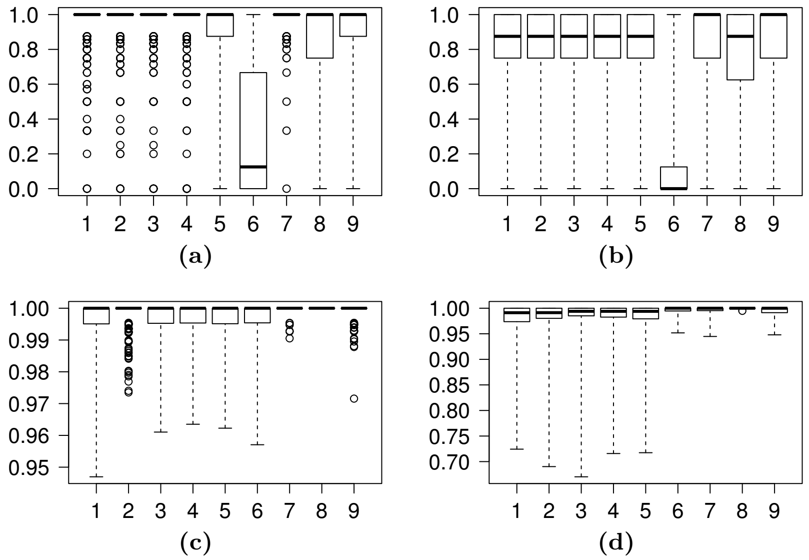
Nr.	Type	N	Kernel	C	Gamma	D	Coef
1	FFNN	6	–	–	–	–	–
2	FFNN	17	–	–	–	–	–
3	FFNN	20	–	–	–	–	–
4	FFNN	25	–	–	–	–	–
5	FFNN	30	–	–	–	–	–
6	SVM	–	Sigmoid	1	0.0001	–	0.01
7	SVM	–	Linear	10	–	–	–
8	SVM	–	Radial	1	0.0001	3	–
9	SVM	–	Polynomial	1	0.1	3	0

Table 7.2: Classifier parametrization with classifier number (Nr.), type, amount of neurons in the hidden layer (N), kernel, cost (C), gamma, degree (D) and coefficient (Coef).

The results of our face recognition show clearly that passing erroneous face detection data to face classifiers strongly decreases the recognition rate. As an on-device implementation of this pan shot face recognition toolchain has to rely on the detected faces only (including erroneous data), this will further strongly decrease the overall performance of the system. We assume that, for our pan shot face unlock approach, our used face detection mechanism will not be sufficient. Hence, more robust and reliable approaches to finding faces in images will are addressed in the succeeding stages.

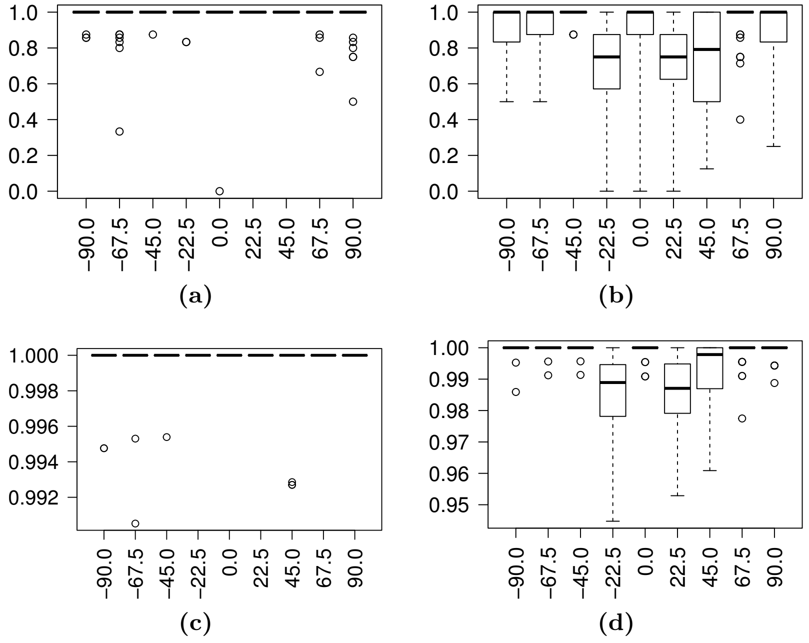
Using correctly detected faces only we achieved a true positive/negative face recognition rate of 97.81%/99.98% and of 86.22%/99.57% with using correctly as well as incorrectly detected faces, for the overall best performing classifier – the SVM with linear kernel. We further analyzed the face recognition rate for different perspectives using this classifier (see figure 7.7).

Interestingly, the results show that, for using correctly detected faces only, there is no remarkable and clearly visible difference in recognition rate from the profile perspectives over the frontal perspective. Therefore, we assume the overall face recognition performance, based on well-known approaches of support vector machines and neural networks, to be sufficient for our further research (using correctly detected faces only). For fine-tuning the ready-made toolchain in a later state of research, more sophisticated approaches to face recognition might still come in use as well.



(a) (b) (c) (d)

Figure 7.6: Face recognition results: true positives for using a) correctly detected faces only, b) incorrectly detected faces as well, and true negatives for using c) correctly detected faces only, d) incorrectly detected faces as well.



(a) (b) (c) (d)

Figure 7.7: Linear kernel SVM face recognition results for different perspectives: true positives for using a) correctly detected faces only, b) incorrectly detected faces as well, and true negatives for using c) correctly detected faces only, d) incorrectly detected faces as well.

Based on these results, we assume a significantly higher difficulty level for tricking the system into authentication when relying on the pan shots instead of only frontal face shots. An attacker would have to replay a synchronized pan shot video stream while moving the attacked device or manufacture a 3D bust of the owner's face. Although we cannot yet quantify the resulting increase in security, we argue that even a small decrease in recognition rate would be outweighed by the increase in security, which would support the day-to-day use of face unlock even for application scenarios with higher security demands.

7.2.2 Discussion

We are using the Viola and Jones algorithm implemented in OpenCV for face detection with cascades optimized for frontal and for side images. We further use support vector machines and neural networks for face recognition, based on previously detected faces. While the approach to face recognition (with mean true positive and true negative rates of above 90%, using correctly detected faces only) seem to be reliable enough for further usage in our pan shot face unlock research, this standard approach to face detection seems to be too error prone (with detection rates down to 60%) – which consequently decreases the face recognition rate as well. Therefore, we address more robust and reliable approaches to finding faces in images from different perspectives in the next stage.

7.3 Stereo Vision Pan Shot Face Recognition Evaluation

In this section we introduce a pan shot face unlock approach based on stereo vision, which was first presented in [63]. Based on improved face recognition, we have now identified Viola and Jones face detection as delivering unsatisfying results, especially when used within the mobile domain. Uneven normalization, false negative and false positive detections decrease the consecutive face recognition results. Therefore, to the pan shot face unlock approach include the usage of range data for face detection and segmentation as well as for subsequent face recognition. We obtain range data from stereo cameras built into the mobile device by using stereo vision algorithms. In preparation to face detection and segmentation, we assemble range templates for multiple perspectives around the user's head. We then perform face detection and segmentation using sliding window template matching and the range template of the corresponding perspective. Finally, we evaluate based on range images contained in the u'smile face database.

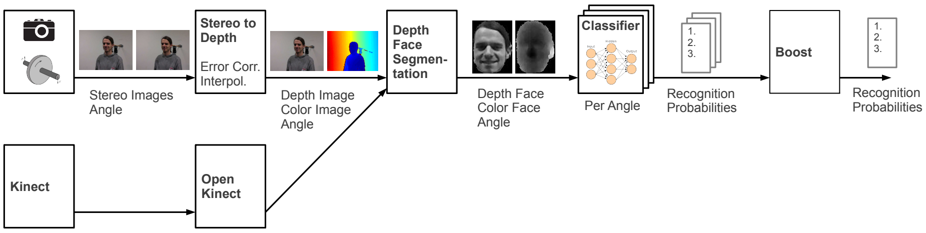


Figure 7.8: Overview of the stereo vision pan shot face unlock system.

7.3.1 Method

For our stereo vision pan shot face unlock (see figure 7.8), we first record grayscale stereo images of the user's head at multiple angles, along with gyroscope sensor data for each pair of images. For a pan shot of 180°, we record nine such image pairs (one pair for about each 22.5°). Using stereo to range algorithms, a range image can be derived from each stereo camera image pair. We use block matching stereo correspondence algorithm implemented in OpenCV [110] as our stereo to range approach – which delivers unsatisfying results (see figure 7.9) the resulting range image has large areas not covered with range information (displayed as white areas). Therefore, the further evaluation of our approach is done on the basis of precalculated range images taken out of our face database, as described in section 6.2.

For processing face images, the face-related information of the image should be cropped first. In our approach, we use range based face segmentation – searching and cutting out faces in a range image – as described in section 7.3.1. Based on segmented grayscale and range face images and their device-angle information at the time of recording, face recognition is performed. For grayscale and range faces and for different perspectives, we use different classifiers. As classifiers, we use Support Vector Machines [143] and Neural Networks [132] for face recognition, as described in section 7.3.1.

Range Face Segmentation

Face recognition should be performed on basis of the grayscale and range input images. To only pass face related data to the classifiers, the face has to be extracted from the image first. One approach to extract a face from an image is to perform grayscale based face detection, such as the well known approach of Viola and Jones [186] with Lienhart and Maydt [115]. In our previous work [65], this approach resulted a) in a notable amount of false positives/negatives, specially for the profile perspective [158], which causes the classifiers to already learn wrong data, and b) in having not face-related information (background) around the corners and borders of the extracted area.

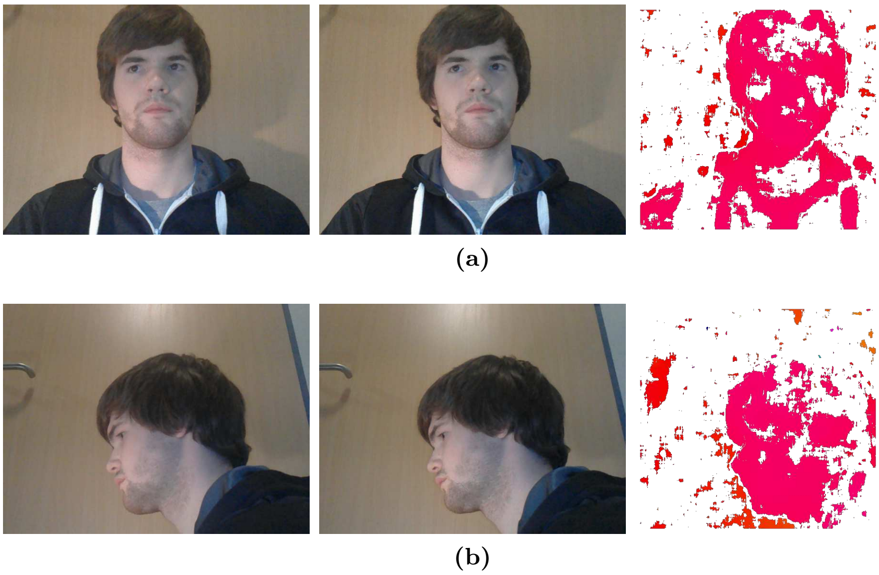


Figure 7.9: Range image created using a mobile device stereo camera image pair from a) frontal and b) profile perspective, using the OpenCV implementation of a block matching algorithm.

Hence, many different approaches for more precise face segmentations have been proposed, such as [126, 151, 167, 168]. For our pan shot face recognition, we rely on a simple, but for our needs yet effective range-template based face segmentation:

1. A coarse person segmentation removes those parts of the image, which have a bigger distance to the camera than a predefined threshold value.
2. The human face is searched in the range image, using an "average human face range template" in combination with a sliding window approach. For each of the nine perspectives there exists on such average human face range template (see figure 7.10).

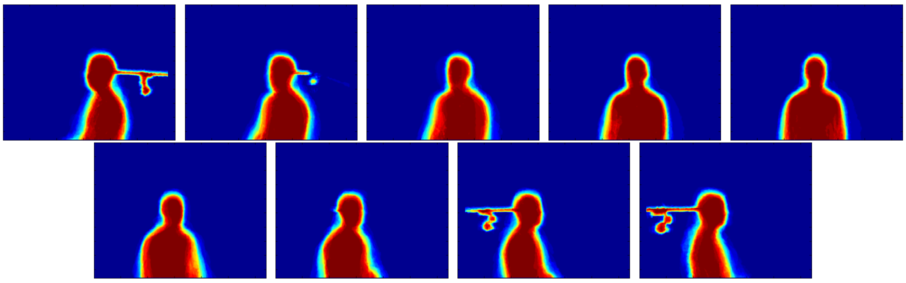
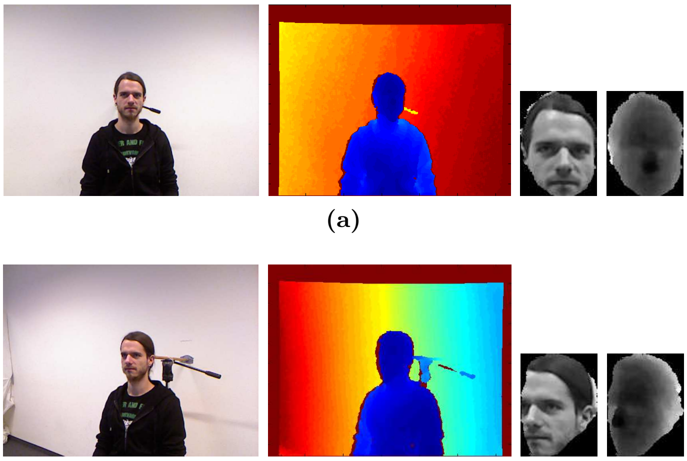


Figure 7.10: Average human range templates for all nine perspectives.

3. Finally, for the best fit of the template in the image, the known area of face in the template is cut out for both the grayscale and the range input image. This results in both one segmented grayscale and range face image (see figure 7.11).

These face segmentation results are not fully accurate, as some minor areas of the faces are missing, and some not face-related information is still included in the extracted faces. Still, the quality of our face segmentation results is good enough for the results to be processed by classifiers, as described in the next section.



(a)

(b)

Figure 7.11: a) Frontal and b) side grayscale and range input images, and their corresponding range based segmented faces.

Face Recognition

Based on the obtained grayscale and range faces from face segmentation, along with the device rotation angle at the time of recording, face recognition is performed. We use different classifiers for grayscale and range images, and for each of the nine perspectives. As face recognition classifiers we again use Support Vector Machines (SVM) [143] and Neural Networks [132]. To find a well adjusted parameter configuration for our recognition, we perform a search for the number of neurones in the hidden layer of the feed forward neural network, and a grid search for the correspondent SVM parameters, as suggested by Hsu et. al. [94].

7.3.2 Test Setup and Results

Training and Test Procedure

The recognition is done as binary classification. For each of the 30 subjects, the face images of the test subject, recorded at a certain angle, represent the positive class, and the images of all other people of the same angle are assigned to the negative class. This leads to the positive class being $\frac{1}{29}$ of the negative class size. The classifiers are trained with 60% of the data of each the positive and negative class (train set). The remaining 40% of the data (test set) are explicitly used to measure the performance of the best classifiers in the end, see section 7.3.2.

Neural Networks: Training for the feed forward neural networks (FFNN) is done as follows: for each angle, subject and classifier, 10 neural networks are trained with the correspondent part of the rtyain set. 30% of the data are used for training the neural networks, 12% percent to perform cross validation to stop the training, and the remaining 18% to evaluate the 10 generated neural networks against each other. The network with the best performance is evaluated using the correspondent part of the train set then.

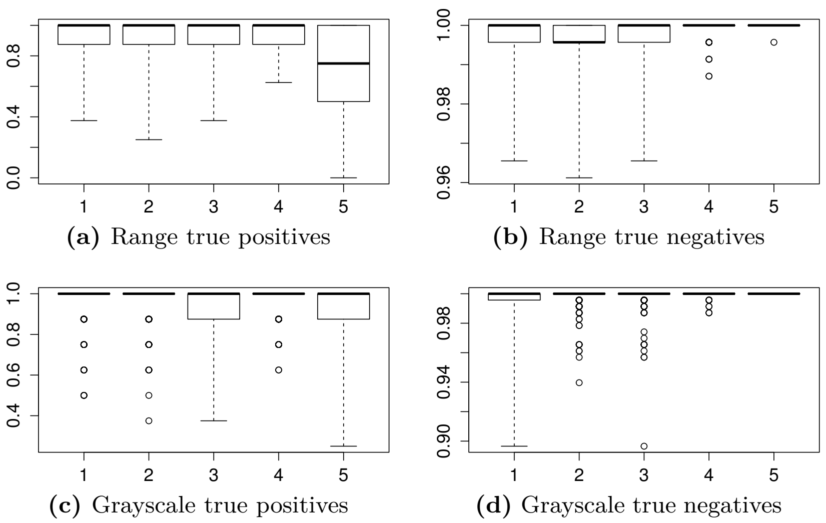
Support Vector Machines: Training for the Support Vector Machines is done as follows: for each angle, subject and classifier, one support vector machine is trained using the correspondent part of the train set, and evaluated on the correspondent part of the test set then.

Recognition Results The classification results of the best performing support vector machine with linear kernel and radial kernel, and best performing neural networks (see table 7.3) are shown in the tables for range and grayscale classification results. The corresponding boxplot provides an overview of true positive and true negative classification results for both range and grayscale faces for all perspectives combined (see figure 7.12).

Nr.	Classifier	Neurons	Kernel	Cost	Gamma
1	FFNN	10	–	–	–
2	FFNN	17	–	–	–
3	FFNN	25	–	–	–
4	SVM	–	Linear	1	–
5	SVM	–	Radial	1	0.01

Table 7.3: Classifier parametrization.

The results show clearly: the range based face recognition performs slightly worse over the grayscale face recognition. For all classifiers – ex-



(a) Range true positives

(b) Range true negatives

(c) Grayscale true positives

(d) Grayscale true negatives

Figure 7.12: Face recognition results, all perspectives combined: for range a) true positives and b) true negatives, and grayscale c) true positives and d) true negatives.

cept of two – the median is 1, but for the mean and first quartile, a clear distinction to the favor of grayscale face recognition is visible: the first quartile of true positive rate for range classifiers is at maximum 87.5%, compared to at least the same value for grayscale classifiers. As the positive class is much smaller in size than the negative class, true negative results are better overall. Again, grayscale performs slightly better than range: the first quartile goes down to 99.57% for three range classifiers, while going down to the same value for one grayscale classifier only. The best performing classifier (SVM with linear kernel) has a mean true positive rate of 93.89% for range faces, which is slightly lower than the mean of 96.85% for grayscale faces. The true negative rate of 99.95% is again slightly lower than the true negative rate of 99.97%. Still, the overall recognition rate obtained by this classifier indicates that range based pan shot face recognition is possible and can be combined with grayscale face recognition results for further usage.

We therefore argue that using additional range faces for pan shot based face unlock will be a feasible approach – even if our range face recognition results are slightly worse over the grayscale recognition results. The slightly worse range recognition rate will be offset by the increased effort, which an attacker will have to accept in order to obtain the additional range data of the user's face.

7.3.3 Discussion

Using range template based detected and segmented faces, we achieve a mean true positive face recognition rate of 93.89% for range and 96.85% for grayscale face images, and a true negative rate of 99.95% for range and 99.97% for grayscale face images. These results indicate that range based face recognition can be used along with grayscale face recognition in a pan shot face unlock scenario. This will increase the amount of required data – and therefore the effort an attacker will have to accept to obtain this data – in order to successfully circumvent a grayscale and range pan shot face unlock system. In the next stage we focus on improving the range template based face detection and segmentation used within our toolchain, with focus on obtaining more precisely segmented faces.

7.4 Improving Range Face Segmentation for Pan Shot Images

In this section we describe improvements to the range template based approach of detecting and segmenting faces, which were first presented in [64]. The improvements include increase precision during range template creation by semi-automatically normalizing face images prior to template creation and increased robustness when matching the template with range images, obtained by an improved matching heuristics. Additionally, we address range images still containing a certain amount of background information in this implementation. We further introduce the optional extension of precisely segmenting a face along its actual contour using GVF snakes – instead of cutting out along the corresponding template border. As for previous implementations, we evaluate our improvements using the grayscale and range face images contained in the u'smile face database.

7.4.1 Method

Our range face detection and segmentation approach (see figure 7.13) performs an initial background removal on input range images. Then, it matches average head range templates of different perspectives to given range images in order to find the most probable head position. As soon as we know the head position, we can already perform an approximate segmentation of the face, using the average head range template contour. As this contour has no possibility to fit the actual, individual face actual contour, we additionally utilize GVF snakes to precisely segment the face in grayscale and range input images. Using segmented faces as input to face recognition, we measure the quality of our detection and segmentation using classifiers for each perspective and test subject.

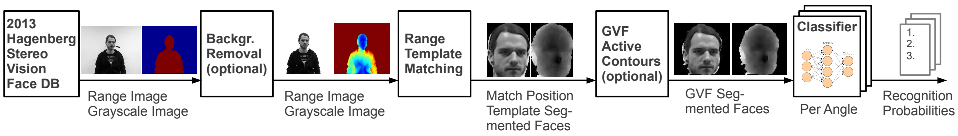


Figure 7.13: Range face segmentation test setup with optional background removal and GVF snake face contour segmentation.

Range Face Template Assembly

This section describes the semi-automatic creation of average head and torso range templates from range images[4]. The template creation toolchain is structured as follows: we first perform a coarse background segmentation to discard information not related to the human face and body. We then normalize the head positions so that they are roughly equal in all images. Finally, we create a) average range images, which represent the average range to the subject, and b) "hit count" templates, which – for each pixel – represent the amount of images in which subjects had range information present.

The template creation is not intended to be performed on the mobile device and has to be done only once for each perspective from which face detection should be performed afterwards.

Coarse Background Segmentation In order to only use range information related to the human head and torso for template creation, we discard all range values bigger than a threshold α. This requires all range images used during template creation to be recorded from roughly the same camera-subject distance, as it is the case with the u'smile face database. Further, α is perspective-dependent: therefore we use histograms of the range value distribution of all images for each perspective to determine the correlating α. The smallest range values (first peak) represent the head and torso, the farthest range values represent background information. Therefore we define α after the first peak (see figure 7.14).

Head Position Normalization After coarse background segmentation, we roughly center the head positions in the range images. Therefore we search the four outer head boundaries and align them along the image center.

To find the top boundary, we search from the image top for the first line containing at least N_t range values. N_t can be adjusted to avoid outliers – e. g. for our implementation we used $N_t = 40$. As this line lies beyond the top

[4] Assuming heads have been normalized to the same size and rotation, as if the images would have been recorded upright and from the same camera-to-subject distance for all participants.

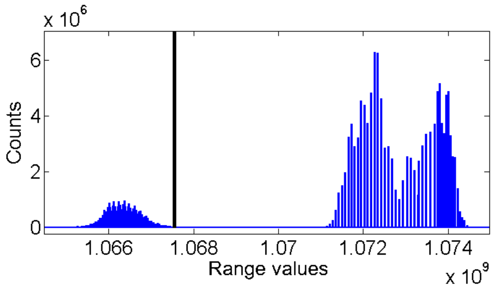
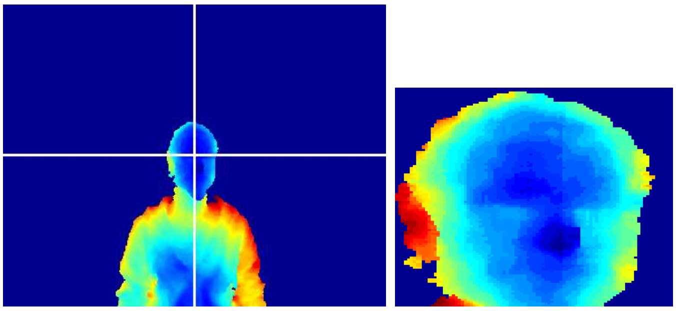


Figure 7.14: Range value distribution of images in the 0° perspective with α marked bolt.

of the head, we go back up 2% of the image height to be sure that all range information is included. The resulting y-coordinate is the head top head boundary. We assume the head bottom boundary lies h pixel beyond the found top boundary, with h being hand-picked from the range $[100, 135]$ depending on the perspective. For frontal perspectives the chin is nearly in the same height as the neck. Therefore a smaller h can be used, as the smallest horizontal area filled with range information is higher than in the portrait perspectives, where the chin is beside the neck and cannot be ignored. To find the right and left boundaries, we now crop the image to the top and bottom boundaries in order to discard range information correlated to the torso. Then we use a similar approach as for finding the top boundary: from the image borders on each side we search for the first column containing at least N_s range values, with $N_s = 70$ in our implementation. Again, we then go back 2% of the image width towards the image borders for including all relevant range information[5]. As we now know the four head boundaries (with the boundaries central point being the head center point), we can a) shift the center points of all heads of a perspective to the same position and b) cutout the heads (see figure 7.15).

Template Assembly Using the head position normalized range images, we can now create four range templates per perspective: a “hit count” torso and face template and an average range torso and face template. The hit count templates represent the amount of images per perspective with range information present at a certain pixel. E. g. as we have at most 620 images per perspective in our test data set, the value range of a hit count template is $[0, 620]$ in our implementation. The average range templates represent the average range information of all images per perspective. We do not consider

[5] As our test data contains an artifact in portrait perspectives, we have to perform an additional artifact removal in our evaluation implementation at this point (see section 7.4.2)



(a) Central position (b) Boundary area

Figure 7.15: Head position normalization with a) marked central point and b) a head cut out by its found boundaries.

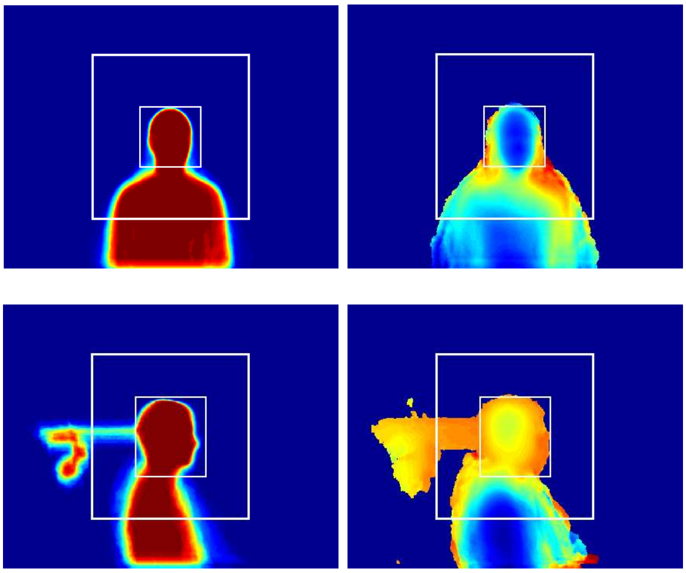
zero values (=background) for the template creation, therefore the range value at a certain pixel is the average of all images only having a foreground value present. The face templates are composed out of the range images cropped to the face areas determined using the head position normalization. The torso images are composed out of the normalized, not cropped range images. We note that template matching conceptually can be performed based on average range templates as well as hit count templates. In our experiments we achieved worse results throughout using the average range templates and therefore describe our approach using the hit count templates only.

We crop the torso templates to a 300 × 300 px area around the users head. These cropped torso templates are used during the first stage of template matching which determines the coarse position of the user's head in a range image. The smaller face templates are used during the second stage of template matching, which aims to improve the accuracy of the face position found during the first stage. Therefore, the second stage template matching is performed in a small area around the initially found face position.

Figure 7.16 shows two examples of head position normalized hit count and average range images. The larger marked region is the 300 × 300 torso template used for coarse matching, the smaller the face template for fine grained matching.

Face Template Matching

We perform face template matching based on sliding window principle and template scaling, so that different face sizes and positions can be matched. Our matching approach further consists of two (basically identical) steps: coarse and fine matching, using the created torso and face range templates. During coarse matching, a rough estimate of the head position is detected



(a) Hit count

(b) Average range

Figure 7.16: Average torso and face images from frontal and profile perspective. Torso and face templates are marked white.

using different scalings and positions of the torso templates. Based on the coarse position we then perform fine matching in the surrounding area using the face templates (using finer sliding window steps with different template scaling and positions again) in order to improve the accuracy of the found face position.

Comparing how well a template matches a certain area in a range image is done using a template matching metric. Normalizing the template and range data depends on the grade of preprocessing applied to the range input data in order for the metric to work correctly. The metric and normalization for range data without background (see section 7.4.1) is applied in case the background has been removed from input range images (as it has been applied to range images used during template creation). In case no background removal has been applied the metric and normalization for range data without background (see section 7.4.1) are applied instead. We implement and evaluate our approach with range data with both background still present and removed (see section 7.4.2).

During our experiments we discovered that searching for correlations between hit count templates and range images is more robust than comparing range values of range images and average range templates. Therefore we only make use of hit count templates subsequent to this point – although average range images might be used as well with different metrics.

Template Matching Metric: without Background This range data and template normalization requires a background-free range image — similar to the images used during template creation. Initially, we convert the hit count template T to the range $[-1, 1]$ using $T = \frac{T - min(T)}{max(T) - min(T)}$. For matching T with an actual range image I we have to normalize I to the range $[-1, 1]$ too, by setting all pixels containing no range information (background) to -1, and all pixels containing range information to 1.

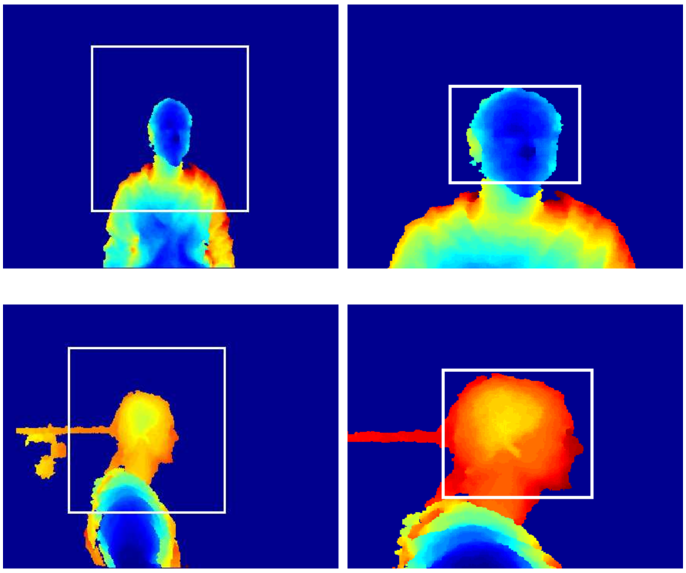
As a) our metric essentially is a multiplication of the normalized hit count template T with the normalized range image I and b) T by now most likely contains more background (-1) than foreground (1) information, the following effect could be observed: as T contains a bigger background than foreground region it will give a bigger weight to matching the background than to matching the foreground. Therefore not matching the background would lead to a stronger decreased metric than not matching the foreground. This could lead to false detections with large background regions being almost perfectly fitted, but the smaller foreground region being missed completely. Consequently we have to correct T before matching, which we did by scaling all values > 0 so that T sums to 0. As side effect the range of T is no longer $[-1, 1]$, but $[-1, N]$ with $N > 1.0$. e. g. for the perspective of 0° we obtained a template range of $[-1, 3.0036]$.

After bringing the range image I and hit count template T to the required range, the metric M can be computed (see equation 7.1). The current template area $A(T)$ is used to normalize M independently of the size of the currently matched area. Regions outside the current size of template T are not taken into account for M. Higher values indicate better matches.

$$M = \frac{1}{A(T)} \sum_{x,y} T_{x,y} \cdot I_{x,y} \tag{7.1}$$

Figure 7.17 shows the best match found by the template matching metric in a range image without background information. The regions marked white are the best matched position with the torso template (big) and face template (small).

Template Matching Metric: with Background In case background removal is not applicable for range images, we use this range data and template normalization which is slightly adapted to work with background information still present in images. Before actually matching template and range image, errors in the range image (regions without information) should be corrected. This is particularly important when matching range images still including background information, as the subsequent application of GVF snakes will likely deliver erroneous results caused by these errors. Especially errors directly at the borders of the face need to be corrected. For the test data used in our evaluation, the real range information of these unknown



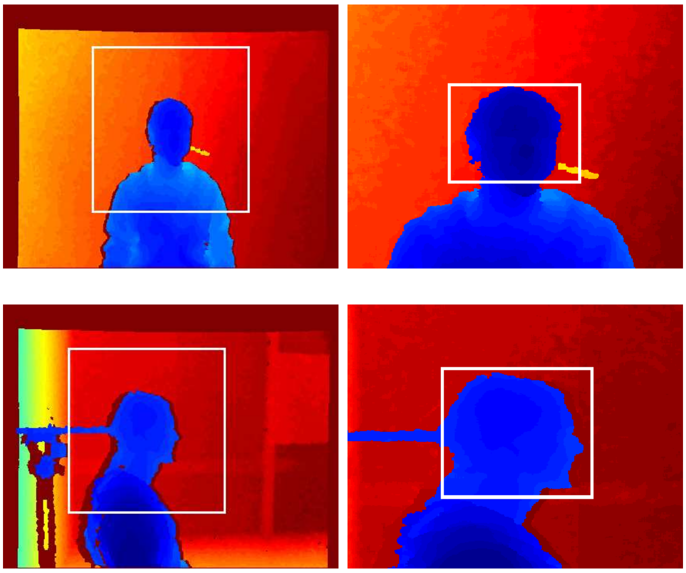
(a) Coarse matching **(b)** Fine matching

Figure 7.17: Best matched face area for coarse and fine matching on range images for a frontal and profile perspective with background information removed

regions next to the face would be "background". Therefore we apply a hole filling algorithm which fills up these unknown regions with background information.

As for matching range images without background, we first normalize the hit count template T to the range $[-1, 1]$, then scale values > 0 so that T sums to 0. As the range image I contains background information, we cannot apply a simple binarization as for matching range not containing background information. Instead we also normalize I to the range $[-1, 1]$. This requires a strong rise in range from the head (foreground) towards the background, so that the foreground will more likely contain positive values, and the background more likely negative values. We then compute the metric using equation 7.1.

Figure 7.18 shows the best match found by the template matching metric on a range image including background information. In comparison to figure 7.17 the results are approximately the same. In case background range values are overall increasing/decreasing towards a certain direction, small shifts along this direction are to be expected.



(a) Coarse matching **(b)** Fine matching

Figure 7.18: Best matched face area for coarse and fine matching on range images for a frontal and profile perspective still containing background information

Sliding Window and Template Scaling We perform face template matching in two stages: coarse matching using the torso template and fine matching using the smaller face template — both from the corresponding perspective. In each stage we perform template matching in a sliding window principle with scaling templates to different sizes.

In the first stage we start with sliding window step sizes S_x and S_y in x- and y-direction – with $S_x = 40px$ and $S_y = 40px$ in our implementation for performance reasons. Within each iteration of this stage, we shift the window by the step size through the image and compute the matching metric using the current position of the sliding window and the template as described in section 7.4.1 and 7.4.1. We append the matching metric result along with template size and position to a list of metric results of this stage. After finishing all iterations of this stage, we know the best match of the template in its current size, given the accuracy defined by current step sizes S_x and S_y. In order to increase the accuracy of the best template match position, we decrease the step sizes and choose a smaller search area around the best four metric positions for the next stage. Additionally, we extend this area with the padding P to ensure potentially best matches at the borders will be included in matching results, with a P = 4px in our implementation.

We further use $\frac{1}{10}$ of the new search area width/height as decreased step size in x/y-direction. We process new stages until the step size is $1px$ or the search area does not change anymore. After finishing all stages, we know the precise position of the best match of the template inside the image – given the current template size.

In order to best match the template in different sizes, we additionally apply an iterative template scaling. For each template size we process template matching as described above and memorize the best matching position, template size and scaling factor. For the start of the template scaling we define three parameters for the scaling range: S_{start}, S_{end} and S_{step}. In our implementation we used $S_{start} = 0.8\%$, $S_{end} = 1.2\%$ and $S_{step} = 0.1\%$. We perform sliding window template matching for all these template sizes and memorize the best matching results for each scaling. We then derive new template scaling factors in order to increase the precision of template scaling. We therefore use the scaling factor S with which the best match was obtained and derive the new scaling parameters as stated in equation 7.2.

$$\begin{aligned} S_{new\_step} &= \frac{S_{step}}{10} \\ S_{new\_start} &= S - 2 \cdot S_{new\_step} \\ S_{new\_end} &= S + 2 \cdot S_{new\_step} \end{aligned} \tag{7.2}$$

We stop template scaling iterations as soon as scaling does not change the window size any more or the matching metric results are the same for all template positions. After finishing sliding window template matching with template scaling we know the match of differently scaled torso templates in the range image in terms of template position and size.

The next step is to search for the exact position of the head inside the torso area. We therefore repeat the process using the head template instead of the torso template and only searching within the area found by torso template matching. In our implementation, we use step sizes of 40px for head template matching – as we did for torso template matching before. After finishing head template matching including different template sizes, we know the head location in the image in terms of rectangular size and position. We crop the image to this area for consecutive face segmentation.

Face Segmentation Approaches

After performing range template based face detection we know the position of the face in the image. As next step we segment this face (discard non-face related information). This can either be done by using the range template contours, or by applying GVF snakes to precisely segment along the individual face contour.

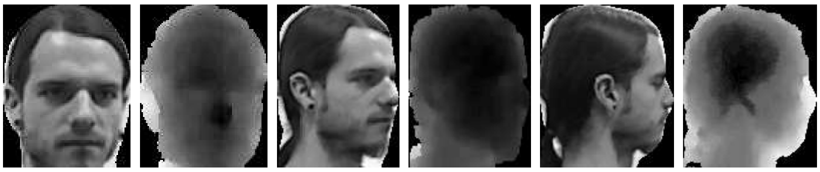


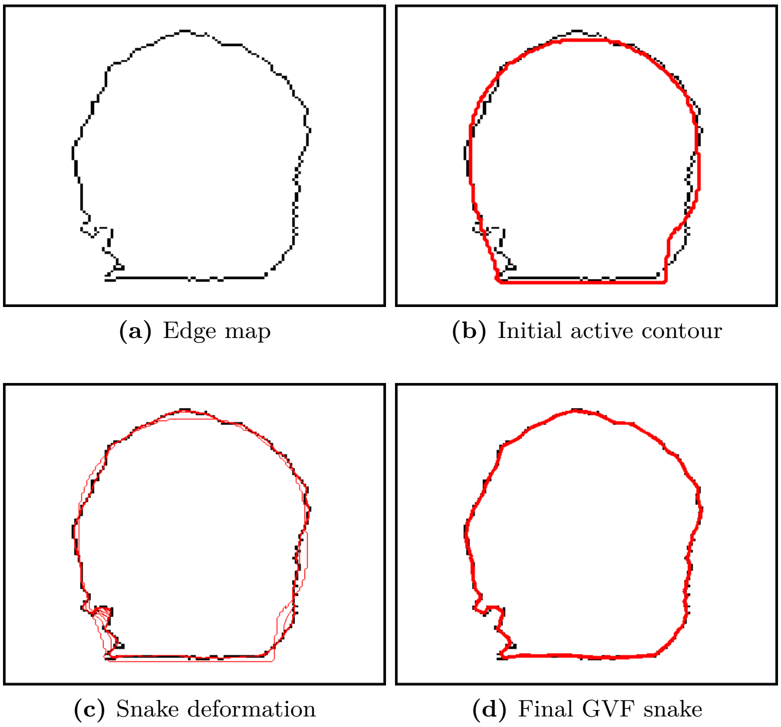
Figure 7.19: Faces cut out by the hit count template contour

Template Segmentation A computationally fast and easy to implement approach which delivers feasible results is to segment the face along the hit count template contour. As the hit count template is usually larger than the detected face (it contains information in all pixels at which at least one range image contained information during template creation), we only consider pixels for which at least $N\%$ of range images contained information. This leads to the hit count template contour getting smaller – and fitting the average face better. Again, N depends on the perspective: for the perspective of 0° we use only hits with at least $N = 50\%$ appearance in all images of this perspective. For the perspective of ±22.5° perspectives we use $N = 60\%$ and for all other perspectives we use $N = 70\%$. When cutting out along the contour of pixels fulfilling the $N\%$ criteria of the hit count template (without using snakes to exactly match the contour), we still can segment faces quite exact (see figure 7.19).

These faces can be used directly as input to face classifiers. Although this approach is faster – as no further segmentation computation is necessary – it has the major drawback of not fitting the actual face contour, as it only takes into account the "average face contour" from the corresponding perspective. Therefore, we additionally use GVF snakes to fit the cut out area precisely to the individual's face contour in the next section.

GVF Snakes Segmentation Snakes are introduced by Kass et. al. [103] as active contour models that create a line towards features – such as edges – based on internal and external constraint forces. Xu and Prince [196, 197] propose gradient vector flow (GVF) as external force for snakes based on a diffusion of the gradient vectors of an feature edge map.

We apply GVF snakes to precisely cut out the face on the individual contour. We position the initial GVF snake on the contour of the preprocessed hit count template from section 7.4.1 – exactly on the contour, on which a cut out using the template only would take place. From there, the GVF snake should evolve towards the face actual contour. For this purpose we first create an edge map of the area around the face in the range image, which was found using template matching [37]. Based on the edge map we calculate the GVF field, acting as external forces which pull the snake towards the edge. As GVF parameters we use a regularization coefficient $\mu = 0.2$ and 80 iteration steps. We further specify the following parameters for the internal GVF snake calculation: $\alpha = 0.05$, $\beta = 0$, $\gamma = 1$ and $\kappa = 0.6$. This results in a trade-off between a quite precise edge matching and fast calculability (see figure 7.20).



(a) Edge map **(b)** Initial active contour

(c) Snake deformation **(d)** Final GVF snake

Figure 7.20: Step-by-step results for precisely fitting a face range contour, from a) edge map up to d) GVF snake.

After performing additional GVF snake segmentation, we naturally obtain more precisely segmented faces, which contain less background information than with cutting out faces at the hit count template contour (see figure 7.21). Again, these faces are used as input to face classifiers in the next section.

To wrap up: in order to perform range template matching we at first assemble templates of the face and torso area from different perspectives. We therefore first remove background in range images, second perform head position normalization and finally create hit count templates from the range images. Before performing template matching, we normalize input images not containing background information slightly differently than range input images still containing background images in order to handle both variants.

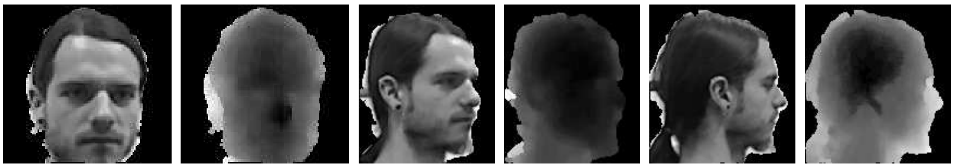


Figure 7.21: Faces segmented by GVF snakes after performing range template matching.

We then perform range face detection based on two staged sliding window template matching and template scaling, using the torso template in the first, the face template in the second stage. In each template matching stage, we recursively reduce the granularity of our search by decreasing the sliding window step width in areas of interest. We do this until we have found the most likely position for the torso in the first stage and the face in the second stage, inside the region marked by the torso.

7.4.2 Test Setup and Results

For evaluation, we implemented our face segmentation approach in Matlab. As test data we use the 2013' Kinect color and range images of the u'smile face database [62]. Using our implementation, we perform face detection and segmentation in 4 different setups:

1. With initial image background removal and without performing GVF snake segmentation.
2. With initial image background removal and with performing GVF snake segmentation.
3. Without initial image background removal and without performing GVF snake segmentation.
4. Without initial image background removal and with performing GVF snake segmentation.

Not performing initial image background removal represents cases in which background removal is not possible for various reasons. Based on the segmented faces we then perform face recognition on range and grayscale images separately and compare our results to previous research [63].

Test Data Artifact Removal

For the image acquisition of the u'smile face database recorded in 2013, a stick behind the head was used to adjust the distance between Kinect sensor and each person. For the head position normalization we need to consider this stick at the back of the head. In the portrait perspectives the side boundary without the stick is found in the same way as for the frontal perspectives. Using this boundary plus a width of 140px the main stick is removed from the image and only the head plus a small additional stick remains. After that we remove the remaining stick by searching the first appearance of 70 range values from the side with the stick, because the remaining stick height is smaller than this threshold and the back of the head boundary is found.

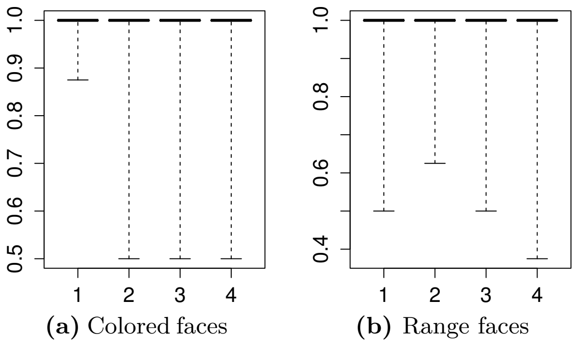
Face Classifiers

Based on segmented faces we perform face recognition on grayscale and range faces separately. Therefore, we treat our face recognition as a binary classification problem. When creating the positive and negative data classes, all images of the particular subject form the positive class, and all images of the other subjects form the negative class. As we have 30 people in our test data, we a) compute 30 such binary classification problems and b) have a negative class being 29 times the size of the positive class. For this reason, the recognition rates for the negative class are expected to be better than those of the positive class – we therefore only state the true positive recognition rates in graphs. We use 60% of the data of each class for training and cross validation. The remaining 40% test data are used exclusively for measuring the final recognition rates. For performing face recognition based on our range template based face segmentation results, we use a Support Vector Machine (SVM) from LibSVM [39] and perform a parameter grid search as suggested by Hsu et. al. [94]. It turns out that the best classifier for our data is a linear SVM with cost of 10.

Results

In comparison to the face detection rate of 77.63% from previous research on the u'smile face database [62], which is based on the OpenCV implementation of Viola and Jones [115, 186], we achieve a 100% correct detection rate with all setups of our range template based approach, as we correctly detect all faces.

The face recognition results (see figure 7.22 and 7.23) clearly show that precise face segmentation using GVF snakes does not improve results (respectively does not improve them significantly). When using initial range image background removal, the average true positive/true negative recognition rates without GVF snakes are 99.78%/100% for color and 98.21%/99.99% for range, compared to 99.33%/100% for color and 98.7%/99.99% for range with GVF snakes. When looking at face segmentation without performing initial background removal, the average true positive/true negative face recognition rates without using GVF snakes are 99.06%/100% for color and 97.4 %/99.98% for range, compared to 98.61%/100% for color and



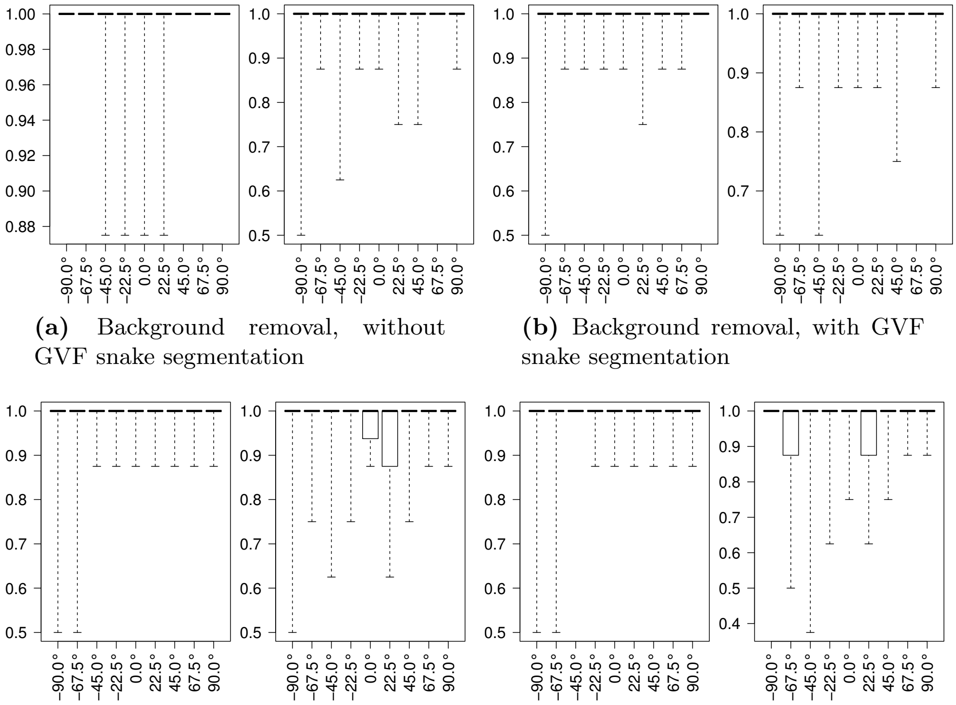
(a) Colored faces **(b)** Range faces

Figure 7.22: Boxplot showing average true positive face recognition rates using segmented faces in a) color and b) range with setup 1-4.

96.82%/99.97% for range when using GVF snakes. Also, using range images is on average less reliable than using grayscale images.

We believe the reason for GVF snakes not improving results further is that a) segmentation based on cutting out the head along the template borders already showed good results and b) the range images contain undefined areas at or next to the real face contour from the start. These range errors lead to an edge map showing a contour not completely matching the real face contour (such as the lower left area in figure 7.20). Therefore, the GVF snake does not completely match the actual face contour, but also includes areas of range error – which vary across the subjects and are learned by face classifiers along with the real face features. We believe that GVF snakes face contour segmentation will result in improved face recognition rates (over a cutout along the template contour) when using more accurate range images in the first place than the Kinect can currently deliver.

Altogether, our current face recognition results are still significantly better than from previous research based on grayscale images only and Viola and Jones face detection with a rectangular crop area [62]. When passing all detected faces (including false positive detections) to the classifiers, the average true positive/negative recognition rate was 86.22%/99.57%. Even when only passing correct detections to the face classifiers, the average true positive/true negative face recognition rate using grayscale images was 97.81%/99.98% – which still is worse than with our current approach. This likely correlates with background information still present in face images and worse normalization in terms of face size and position for the previous approach. We further achieve improved results compared to previous research also based on range template matching [63], where the average true positive/true negative recognition rate was 96.85%/99.97% for color and 93.89%/99.95% for range images – with the most important improvement being the precise data normalization during template creation and matching.



(a) Background removal, without GVF snake segmentation

(b) Background removal, with GVF snake segmentation

(c) Background removal, without GVF snake segmentation

(d) Background removal, with GVF snake segmentation

Figure 7.23: Boxplot showing true positive face recognition results separated for perspectives, using segmented faces in color (left) and range (right).

7.4.3 Discussion

Our results indicate that face detection and segmentation based on range information might be a very effective approach in general for finding single faces in a mobile device unlock scenario. Using range template matching, we achieve an error free face detection on Kinect color and range images of the u'smile face database. Faces segmented by our approach are normalized in position and size and contain little background information. As both of these are important to face recognition, we naturally achieve better face recognition results compared to previous research using the same data. Using a linear Support Vector Machine as face classifier we achieve average true positive recognition rates above 98% on grayscale and 96% on range images from our test set.

In order to apply range template based face detection and segmentation in the mobile domain, mobile devices must be capable of taking such range images, e. g. with stereo cameras. Some current smartphones already contain such cameras mounted on the back side – for more convenient usage they would be required to contain stereo cameras on the front side too. When using stereo cameras, a further prerequisite to successfully performing a range based face unlock in the mobile domain are computationally fast and robust stereo-to-range algorithms that generate range images with only a small amount of erroneous areas. As many existing stereo vision algorithms are either computationally too intensive or deliver inadequate results when applied on data recorded with typical mobile device quality, there is a strong need for improved stereo vision algorithms applicable in the mobile domain as necessary groundwork to successful mobile device stereo vision face unlock.

Chapter 8

Conclusion

We are working on a mobile device unlocking approach which uses all data available from recording a mobile device pan shot around the user's head. Our approach is intended to on the one hand increase the level of security which is realistically applied in practice during unlocking, while on the other hand retaining high usability due to fast usage and not requiring the user to remember an unlocking secret. In comparison to classical mobile device unlocking approaches, our approach is intended to be shoulder surfing resistant by design – as it is an inherence based approach in contrast to a knowledge based approach. In comparison to face unlocking approaches using frontal face information only, our approach is designed to be harder to circumvent using photo attacks – as an attacker would be required to obtain facial information from multiple perspectives, such as obtaining a 3D model of the user's head. In order to develop a high quality toolchain to conduct a pan shot based mobile device unlock, we proposed our approach in four stages, each containing an implementation and evaluation in order identify weaknesses and improve the approach within the next stage.

Our initial implementation uses a smart phone with gyroscope sensor and built-in camera in order to evaluate feasibility of a mobile device pan shot face unlock. The pan shot data aggregation is based on grayscale images and gyroscope data in order to distinguish between different perspectives. For face detection we utilized Viola and Jones frontal and profile Haar-cascades in order to detect faces from all perspectives from which images have been recorded. For face recognition we utilized Eigenfaces classifiers – one classifier per perspective, with seven perspectives in total. Within our evaluation we mainly identify Eigenfaces for recognition as being to unreliable for further usage in mobile domain face recognition. For our second implementation, we therefore exchange Eigenfaces for recognition with using neural networks and support vector machines for face recognition, but keep Viola and Jones based face detection. Again, we use one face recognition classifier per perspective, so that pan shot data is classified by nine classifiers in total. We further evaluate our pan shot face unlock toolchain with a larger test set. Within this second implementation, we identify our face recognition approach to be sufficient for further usage in future research, but identify using Viola and Jones face detection as being to unreliable for usage in a mobile device pan shot scenario – especially because of a) many incorrect detections, b) segmented faces being normalized unequally and containing a variable amount of background information and c) only distinguishing between frontal and profile face detection instead of all nine used perspectives. For our third implementation, we consequently develop a novel face detection approach for mobile device pan shot face unlock, which is based on range images and is intended to work from any perspective. With using range images, we now require a mobile device with stereo camera and a gyroscope sensor in order to derive range images from using stereo vision. We create range templates from multiple perspectives and match pan shot images with the template of the corresponding perspective to detect and segment the user's face. We keep neural networks and support vector machines as face recognition classifiers to evaluate our approach. Results indicate that using range information for finding and segmenting a single face might in general be a good approach – but we also identify the template creation and matching process requiring further investigation in order to improve the result quality. Consequently, for our fourth and last implementation we focus on improving the pan shot face detection and segmentation approach based on range images. We semi-automatically normalize range images for template creation. We further evaluate improvements based on prise face segmentation using snakes – which turn out to not improve results over template border cutout based segmentation, as range images are too erroneous for applying snakes themselves.

Therefore, one point left open by the implementations of this thesis is the usage of stereo vision algorithms applicable within a stereo vision mobile device pan shot unlock scenario. The stereo vision approaches utilized on the mobile device during the implementation were insufficient, as they delivered too erroneous information for usage in facial recognition tasks. Other, openly available stereo vision algorithms are often computationally intensive (but could possible be accelerated using specialized hardware in mobile devices). Therefore, future research will – besides others – need to comprehensively evaluate novel stereo vision approaches for feasibility within our approach. Besides focusing on stereo vision algorithms, the developed mobile device pan shot face unlock toolchain (using range images for range template matching based face detection and segmentation, support vector machines and neural networks for face recognition and majority voting for obtaining a scalar recognition probability for unlocking/not unlocking the device) requires further, extensive tests, with no longer using ideal/limited test set recording conditions. These tests need to be conducted using data actually recorded with mobile devices to obtain the large variance (in terms of e. g. illumination, image quality, changed style) typical for data recorded with mobile devices. Consequently, an extension of the u'smile face database by large data sets recorded with mobile devices in very diverse situation is the logical next step for evaluation and eventual improvements to the current pan shot unlock toolchain. Another point left open by our current approach and implementation is the device automatically recognizing users wanting to start and end a pan shot by pointing the camera/cameras of their mobile device towards one ear. Future research might address this issue – but as it is relatively easy for users to press a button on the mobile device in order to start/end a pan shot, this is considered to be a minor issue.

By now, our face detection and recognition does not check if a recorded image actually contains a face before processing it. In case of an image being presented to the camera which does not contain a face at all, the area most likely containing a face will be segmented and handed to face recognition nevertheless. We did not quantify the security impact of such images being presented for authentication yet as they will likely be classified as originated by the negative class, but future work will need to incorporate a check for if an image actually contains a face before processing it within face recognition. This could be done e. g. by adding non-face images to negative class used for training face classifiers or with using a separate classifier deciding upon an image area segmented after face detection actually contains a face.

Further, face classifiers of different perspectives do not check for pan shot data actually being originated by the same pan shot. Therefore, an attacker would conceptually be able to conduct a photo attack by presenting images to the camera showing the authorized user from correct perspectives – but which were recorded in totally different situations. Such a pan shot photo attack could be based on obtaining a large amount of images of the authorized user (e. g. by fetching them from a social network) and then selecting images showing the user from the required perspectives to obtain a "pan shot" image series. Further, the attacker would need to rotate the device while (automatically) presenting the correlated images to the camera – which will be the easiest part of the attack. This attack can be prevented by checking that images used for different perspectives were actually recorded within the same pan shot, or e. g. by recording a video stream instead of single images and a check for drastic changes within frames changes.

Another potential attack to pan shot face unlock is based on deriving a 3D head model from images showing the authorized user's face from different perspectives (as done recently in [19] with using a single image using FaceGen modeler[1]). At least with only using grayscale information during the face recognition process, we expect this head model attack to have a huge potential. Texture information extracted from images could be projected onto an arbitrary head model, which could then e. g. be displayed in rota-

[1] http://www.facegen.com/modeller.htm

tion on another mobile device and used for conducting a head model attack. We expect our approach to be less prone to the head model attack when using range information – as range face information a) cannot be obtained as easily as grayscale face information and b) deriving exact range information would require a huge amount of grayscale images or video streams from multiple perspectives recorded with high quality.

In order to increase face recognition accuracy, bagging and boosting of classifiers from same perspectives might be in the focus of future work too. Besides the challenges especially present in the mobile domain, such as large diversity of illumination conditions or bad image quality, there exist further challenges currently focused by non-mobile face recognition – such as distinguishing between identical twins [144]. These challenges also apply for our mobile device unlocking scenario, and therefore will have to be addressed at some point in the future too.

References

Literature

[1] Andrea F. Abate et al. “2D and 3D face recognition: A survey”. In: *Pattern Recognition Letters* 28.14 (Oct. 2007), pp. 1885–1906. URL: http://dx.doi.org/10.1016/j.patrec.2006.12.018 (cit. on p. 27).

[2] A.E. Abdel-Hakim and M. El-Saban. “Face authentication using graph-based low-rank representation of facial local structures for mobile vision applications”. In: *Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on*. 2011, pp. 40–47 (cit. on p. 16).

[3] R. Abiantun and M. Savvides. “Dynamic three-bin real AdaBoost using biased classifiers: An application in face detection”. In: *IEEE 3rd International Conference on Biometrics: Theory, Applications, and Systems (BTAS ’09)*. Sept. 2009, pp. 1–6 (cit. on p. 21).

[4] Timo Ahonen, Abdenour Hadid, and Matti Pietikäinen. “Face Recognition with Local Binary Patterns”. In: *8th European Conference on Computer Vision (ECCV 2004)*. Ed. by Tomás Pajdla and Jiri Matas. Vol. 3021. Lecture Notes in Computer Science. Springer, May 2004, pp. 469–481 (cit. on p. 26).

[5] M. A. Aizerman, E. A. Braverman, and L. Rozonoer. “Theoretical foundations of the potential function method in pattern recognition learning”. In: *Automation and Remote Control*. Automation and Remote Control 25. 1964, pp. 821–837 (cit. on p. 32).

[6] Lale Akarun, Berk Gokberk, and Albert Ali Salah. “3D Face Recognition for Biometric Applications”. In: *Proceedings of the 13th European Signal Processing Conference (EUSIPCO)*. Antalya, Sept. 2005 (cit. on p. 27).

[7] A. Anjos and S. Marcel. “Counter-measures to photo attacks in face recognition: A public database and a baseline”. In: *International Joint Conference on Biometrics (IJCB 2011)*. 2011, pp. 1–7 (cit. on p. 14).

[8] Adam J. Aviv et al. “Practicality of accelerometer side channels on smartphones”. In: *Proceedings of the 28th Annual Computer Security Applications Conference*. ACSAC ’12. Orlando, Florida: ACM, 2012, pp. 41–50. URL: http://doi.acm.org/10.1145/2420950.2420957 (cit. on p. 9).

[9] Adam J. Aviv et al. “Smudge attacks on smartphone touch screens”. In: *Proceedings of the 4th USENIX conference on offensive technologies*. Washington, DC, 2010, pp. 1–7. URL: http://dl.acm.org/citation.cfm?id=1925004.1925009 (cit. on p. 8).

[10] M. Baloul, E. Cherrier, and C. Rosenberger. “Challenge-based speaker recognition for mobile authentication”. In: *Biometrics Special Interest Group (BIOSIG), 2012 BIOSIG - Proceedings of the International Conference of the*. 2012, pp. 1–7 (cit. on p. 10).

[11] Wei Bao et al. “A liveness detection method for face recognition based on optical flow field”. In: *International Conference on Image Analysis and Signal Processing (IASP 2009)*. Apr. 2009, pp. 233–236 (cit. on p. 14).

[12] J. Batlle, E. Mouaddib, and J. Salvi. “Recent progress in coded structured light as a technique to solve the correspondence problem: a survey”. In: *Pattern Recognition* 31.7 (1998), pp. 963–982. URL: http://www.sciencedirect.com/science/article/pii/S0031320397000745 (cit. on p. 28).

[13] Peter N. Belhumeur, João P. Hespanha, and David J. Kriegman. “Eigenfaces vs. Fisherfaces: Recognition Using Class Specific Linear Projection”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 19.7 (July 1997), pp. 711–720 (cit. on pp. 25, 59).

[14] Noam Ben-Asher et al. “On the need for different security methods on mobile phones”. In: *Proceedings of the 13th International Conference on Human Computer Interaction with Mobile Devices and Services*. MobileHCI ’11. Stockholm, Sweden: ACM, 2011, pp. 465–473. URL: http://doi.acm.org/10.1145/2037373.2037442 (cit. on p. 3).

[15] Samarth Bharadwaj et al. “Computationally Efficient Face Spoofing Detection with Motion Magnification”. In: *The IEEE Computer Society Workshop on Biometrics*. June 2013 (cit. on p. 14).

[16] Robert Biddle, Sonia Chiasson, and P.C. Van Oorschot. “Graphical passwords: Learning from the first twelve years”. In: *ACM Comput* 44.4 (Sept. 2012), 19:1–19:41. URL: http://doi.acm.org/10.1145/2333112.2333114 (cit. on p. 8).

[17] Christopher M. Bishop. *Neural Networks for Pattern Recognition*. New York, NY, USA: Oxford University Press, Inc., 1995 (cit. on p. 34).

[18] Christopher M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2006 (cit. on p. 31).

[19] Arman Boehm et al. “SAFE: Secure Authentication with Face and Eyes”. In: *IEEE PRISMS 2013*. Atlantic City, June 2013 (cit. on p. 89).

[20] J. Bonneau. “The Science of Guessing: Analyzing an Anonymized Corpus of 70 Million Passwords”. In: *Security and Privacy (SP), 2012 IEEE Symposium on*. 2012, pp. 538–552 (cit. on p. 2).

[21] Bernhard E. Boser, Isabelle M. Guyon, and Vladimir N. Vapnik. “A training algorithm for optimal margin classifiers”. In: *Proceedings of the fifth annual workshop on Computational learning theory*. COLT ’92. Pittsburgh, Pennsylvania, USA: ACM, 1992, pp. 144–152. URL: http://doi.acm.org/10.1145/130385.130401 (cit. on p. 32).

[22] Fabrice Bourel et al. “Robust Facial Feature Tracking”. In: *Proc. 11th British Machine Vision Conference*. 2000, pp. 232–241 (cit. on p. 25).

[23] Kevin W. Bowyer, Kyong Chang, and Patrick Flynn. “A survey of approaches and challenges in 3D and multi-modal 3D + 2D face recognition”. In: *Computer Vision and Image Understanding* 101.1 (Jan. 2006), pp. 1–15. URL: http://dx.doi.org/10.1016/j.cviu.2005.05.005 (cit. on p. 27).

[24] K.L. Boyer and A.C. Kak. “Color-Encoded Structured Light for Rapid Active Ranging”. In: *Pattern Analysis and Machine Intelligence, IEEE Transactions on* IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI-9).1 (1987), pp. 14–28 (cit. on p. 28).

[25] G. Bradski. “The OpenCV Library”. In: *Dr. Dobb’s Journal of Software Tools* (2000) (cit. on p. 54).

[26] Gary Bradski and Adrian Kaehler. *Learning OpenCV: Computer Vision with the OpenCV Library*. Cambridge, MA: O’Reilly, 2008 (cit. on p. 29).

[27] H. Bredin et al. “Detecting Replay Attacks in Audiovisual Identity Verification”. In: *Acoustics, Speech and Signal Processing, 2006. ICASSP 2006 Proceedings. 2006 IEEE International Conference on*. Vol. 1. May 2006, p. I (cit. on p. 14).

[28] Leo Breiman. “Arcing classifiers”. In: *The Annals of Statistics* 26.3 (1998), pp. 801–849 (cit. on p. 37).

[29] Leo Breiman. “Bagging predictors”. In: *Machine Learning* 24.2 (Aug. 1996), pp. 123–140. URL: http://dx.doi.org/10.1023/A:1018054314350 (cit. on p. 37).

[30] Alexander M. Bronstein, Michael M. Bronstein, and Ron Kimmel. “Expression-invariant 3D face recognition”. In: *Proceedings of the 4th international conference on Audio- and video-based biometric person authentication*. AVBPA’03. Guildford, UK: Springer-Verlag, 2003, pp. 62–70. URL: http://dl.acm.org/citation.cfm?id=1762222.1762232 (cit. on p. 26).

[31] Alan Brooks and Li Gao. *Eigenface and Fisherface Performance Across Pose*. ECE 432 Computer Vision with Professor Ying Wu Final Project Report. June 2004. URL: http://dailyburrito.com/projects/facerecog/FaceRecReport.html (cit. on p. 25).

[32] R. Brunelli and T. Poggio. “Face recognition: features versus templates”. In: *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 15.10 (Oct. 1993), pp. 1042–1052 (cit. on p. 25).

[33] R. Brunelli and T. Poggio. “Face recognition through geometrical features”. In: *European Conference on Computer Vision (ECCV’92)*. Ed. by G. Sandini. Vol. 588. Lecture Notes in Computer Science. Springer Berlin / Heidelberg, 1992, pp. 792–800. URL: http://dx.doi.org/10.1007/3-540-55426-2_90 (cit. on p. 25).

[34] Christopher J. C. Burges. “A Tutorial on Support Vector Machines for Pattern Recognition”. In: *Data Mining and Knowledge Discovery* 2.2 (June 1998), pp. 121–167. URL: http://dx.doi.org/10.1023/A:1009715923555 (cit. on p. 32).

[35] Liang Cai and Hao Chen. “On the practicality of motion based keystroke inference attack”. In: *Proceedings of the 5th international conference on Trust and Trustworthy Computing*. TRUST’12. Vienna, Austria: Springer-Verlag, 2012, pp. 273–290. URL: http://dx.doi.org/10.1007/978-3-642-30921-2_16 (cit. on pp. 8, 9).

[36] Liang Cai and Hao Chen. “TouchLogger: inferring keystrokes on touch screen from smartphone motion”. In: *Proceedings of the 6th USENIX conference on Hot topics in security*. HotSec’11. San Francisco, CA: USENIX Association, 2011, pp. 9–9. URL: http://dl.acm.org/citation.cfm?id=2028040.2028049 (cit. on p. 9).

[37] Joh Canny. “A Computational Approach to Edge Detection”. In: *Pattern Analysis and Machine Intelligence, IEEE Transactions on* PAMI-8.6 (1986), pp. 679–698 (cit. on p. 80).

[38] D. Caspi, N. Kiryati, and Joseph Shamir. “Range imaging with adaptive color structured light”. In: *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 20.5 (1998), pp. 470–480 (cit. on p. 28).

[39] Chih-Chung Chang and Chih-Jen Lin. "LIBSVM: A library for support vector machines". In: *ACM Transactions on Intelligent Systems and Technology* 2.3 (3 Apr. 2011). Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm, pp. 1–27 (cit. on pp. 62, 83).

[40] Kyong I. Chang, Kevin W. Bowyer, and Patrick J. Flynn. "Face recognition using 2D and 3D facial data". In: *ACM Workshop on Multimodal User Authentication*. 2003, pp. 25–32 (cit. on p. 27).

[41] R. Chellappa, C.L. Wilson, and S. Sirohey. "Human and machine recognition of faces: a survey". In: *Proceedings of the IEEE* 83.5 (1995), pp. 705–741 (cit. on p. 27).

[42] I-Kuei Chen et al. "A real-time multi-user face unlock system via fast sparse coding approximation". In: *Consumer Electronics – Berlin (ICCE-Berlin), 2012 IEEE International Conference on*. Jan. 2012, (cit. on p. 16).

[43] Girija Chetty and Michael Wagner. "Robust face-voice based speaker identity verification using multilevel fusion". In: *Image and Vision Computing* 26.9 (2008), pp. 1249–1260. URL: http://www.sciencedirect.com/science/article/pii/S0262885608000498 (cit. on p. 10).

[44] Ivana Chingovska, André Anjos, and Sébastien Marcel. "On the Effectiveness of Local Binary Patterns in Face Anti-spoofing". In: *Proceedings of the 11th International Conference of the Biometrics Special Interes Group*. Sept. 2012 (cit. on p. 14).

[45] N.L. Clarke and S.M. Furnell. "Authentication of users on mobile telephones – A survey of attitudes and practices". In: *Computers and Security* 24.7 (2005), pp. 519–527. URL: http://www.sciencedirect.com/science/article/pii/S0167404805001446 (cit. on p. 3).

[46] Michael Collins and Yoram Singer. "Unsupervised Models for Named Entity Classification". In: *In Proceedings of the Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora*. 1999, pp. 100–110 (cit. on p. 36).

[47] T. F. Cootes et al. "View-Based Active Appearance Models". In: *Image and Vision Computing* 20 (2002), pp. 657–664 (cit. on p. 25).

[48] Lorrie Cranor and Simson Garfinkel. *Security and Usability*. O'Reilly Media, Inc., 2005 (cit. on p. 3).

[49] Nello Cristianini and John Shawe-Taylor. *An introduction to support Vector Machines: and other kernel-based learning methods*. New York, NY, USA: Cambridge University Press, 2000 (cit. on pp. 32, 34).

[50] Navneet Dalal and Bill Triggs. "Histograms of Oriented Gradients for Human Detection". In: *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05).* Vol. 1. CVPR '05. Washington, DC, USA: IEEE Computer Society, 2005, pp. 886–893. URL: http://dx.doi.org/10.1109/CVPR.2005.177 (cit. on p. 21).

[51] Nikolay Degtyarev and Oleg Seredin. "Comparative testing of face detection algorithms". In: *Proceedings of the 4th international conference on Image and signal processing.* ICISP'10. Trois-Rivi&#232;res, QC, Canada: Springer-Verlag, 2010, pp. 200–209. URL: http://dl.acm.org/citation.cfm?id=1875769.1875796 (cit. on p. 22).

[52] K. Delac and M. Grgic. "A survey of biometric recognition methods". In: *Electronics in Marine, 2004. Proceedings Elmar 2004. 46th International Symposium.* 2004, pp. 184–193 (cit. on p. 9).

[53] Alexander De Luca, Katja Hertzschuch, and Heinrich Hussmann. "ColorPIN: securing PIN entry through indirect input". In: *CHI '10: Proceedings of the 28th international conference on Human factors in computing systems.* New York, NY, USA: ACM, 2010, pp. 1103–1106 (cit. on p. 7).

[54] Alexander De Luca et al. "Back-of-device authentication on smartphones". In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems.* CHI '13. New York, NY, USA: ACM, 2013, pp. 2389–2398. URL: http://doi.acm.org/10.1145/2470654.2481330 (cit. on p. 8).

[55] Alexander De Luca et al. "Touch me once and i know it's you! Implicit authentication based on touch screen patterns". In: *Proceedings of the 2012 ACM annual conference on Human Factors in Computing Systems.* CHI '12. New York, NY, USA: ACM, 2012, pp. 987–996. URL: http://doi.acm.org/10.1145/2208516.2208544 (cit. on p. 8).

[56] U.R. Dhond and J.K. Aggarwal. "Structure from stereo-a review". In: *Systems, Man and Cybernetics, IEEE Transactions on* 19.6 (1989), pp. 1489–1510 (cit. on p. 29).

[57] ThomasG. Dietterich. "An Experimental Comparison of Three Methods for Constructing Ensembles of Decision Trees: Bagging, Boosting, and Randomization". English. In: *Machine Learning* 40.2 (2000), pp. 139–157. URL: http://dx.doi.org/10.1023/A%3A1007607513941 (cit. on p. 37).

[58] Damien Douxchamps and Nick Campbell. "Robust Real Time Face Tracking for the Analysis of Human Behaviour". In: *Machine Learning for Multimodal Interaction.* Ed. by Andrei Popescu-Belis, Steve Renals, and HervÃ© Bourlard. Vol. 4892. Lecture Notes in Computer

Science. Berlin / Heidelberg: Springer, 2008, pp. 1–10 (cit. on pp. 21, 58, 61).

[59] Stephan Dreiseitl. *Lecture Notes Machine Learning, University of Applied Sciences Upper Austria.* 2013 (cit. on p. 32).

[60] Hossein Falaki et al. "Diversity in smartphone usage". In: *The 8th Annual International Conference on Mobile Systems, Applications, and Services (MobiSys 2010).* 2010, pp. 179–194 (cit. on p. 1).

[61] N. Fatima and T.F. Zheng. "Short Utterance Speaker Recognition A research Agenda". In: *International Conference on Systems and Informatics (ICSAI) 2012.* 2012, pp. 1746–1750 (cit. on p. 10).

[62] Rainhard Dieter Findling and Rene Mayrhofer. "Towards Pan Shot Face Unlock: Using Biometric Face Information from Different Perspectives to Unlock Mobile Devices". In: *International Journal of Pervasive Computing and Communications* 9.3 (2013). *(accepted for publication)* (cit. on pp. vii, 23, 60, 82–84).

[63] Rainhard Dieter Findling and Rene Mayrhofer. "Towards Secure Personal Device Unlock using Stereo Camera Pan Shots". In: *Second International Workshop on Mobile Computing Platforms and Technologies (MCPT 2013).* Ed. by Rene Mayrhofer and Clemens Holzmann. *(accepted for publication).* 2013 (cit. on pp. vii, 41, 48, 65, 82, 84).

[64] Rainhard Dieter Findling et al. "Range Face Segmentation: Face Detection and Segmentation for Authentication in Mobile Device Range Images". In: *Proc. MoMM 2013: 11th International Conference on Advances in Mobile Computing and Multimedia.* Ed. by Ismail Khalil. *submitted for review.* New York, NY, USA: ACM, Dec. 2013 (cit. on pp. vii, 19, 71).

[65] Rainhard Findling and Rene Mayrhofer. "Towards Face Unlock: On the Difficulty of Reliably Detecting Faces on Mobile Phones". In: *Proc. MoMM 2012: 10th International Conference on Advances in Mobile Computing and Multimedia.* Ed. by Ismail Khalil. Bali, Indonesia: ACM, Dec. 3–5, 2012, pp. 275–280 (cit. on pp. vii, 52, 62, 66).

[66] David Fofi, Tadeusz Sliwa, and Yvon Voisin. "A comparative survey on invisible structured light". In: *Proceedings of SPIE, Machine Vision Applications in Industrial Inspection XII* 5303 (May 2004), pp. 90–98. URL: http://dx.doi.org/10.1117/12.525369 (cit. on p. 28).

[67] Yoav Freund. "An Adaptive Version of the Boost by Majority Algorithm". English. In: *Machine Learning* 43.3 (2001), pp. 293–318. URL: http://dx.doi.org/10.1023/A%3A1010852229904 (cit. on p. 36).

[68] Yoav Freund and Robert E. Schapire. "Experiments with a New Boosting Algorithm". In: *International Conference on Machine Learning*. 1996, pp. 148–156. URL: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.51.6252 (cit. on p. 36).

[69] Yoav Freund et al. "An efficient boosting algorithm for combining preferences". In: *Journal of Machine Learning Research* 4 (Dec. 2003), pp. 933–969. URL: http://dl.acm.org/citation.cfm?id=945365.964285 (cit. on p. 36).

[70] Y. Freund and R. E. Schapire. "A Decision Theoretic Generalization of On-Line Learning and an Application to Boosting". In: *Second European Conference on Computational Learning Theory*. Ed. by Paul M. B. Vitányi. Aix-en-Provence, France, 1995, pp. 23–37. URL: citeseer.nj.nec.com/freund95decisiontheoretic.html (cit. on p. 36).

[71] Jerome H. Friedman. "Greedy Function Approximation: A gradient boosting machine". In: *The Annals of Statistics* 29.5 (2001), pp. 1189–1232 (cit. on p. 36).

[72] Jerome H. Friedman. "Stochastic gradient boosting". In: *Computational Statistics and Data Analysis* 38.4 (Feb. 2002), pp. 367–378. URL: http://dx.doi.org/10.1016/S0167-9473(01)00065-2 (cit. on p. 36).

[73] Jerome Friedman, Trevor Hastie, and Robert Tibshirani. "Additive Logistic Regression: a Statistical View of Boosting". In: *Annals of Statistics* 28 (1998), p. 2000 (cit. on p. 36).

[74] Stephen Fried. *Mobile Device Security: A Comprehensive Guide to Securing Your Information in a Moving World*. 1st. Boston, MA, USA: Auerbach Publications, 2010 (cit. on p. 1).

[75] R.W. Frischholz and A. Werner. "Avoiding replay-attacks in a face recognition system using head-pose estimation". In: *IEEE International Workshop on Analysis and Modeling of Faces and Gestures (AMFG 2003)*. Oct. 2003, pp. 234–235 (cit. on p. 14).

[76] Steven Furnell, Nathan Clarke, and Sevasti Karatzouni. "Beyond the PIN: Enhancing user authentication for mobile devices". In: *Computer Fraud and Security* 2008.8 (2008), pp. 12–17. URL: http://www.sciencedirect.com/science/article/pii/S1361372308701271 (cit. on p. 1).

[77] Yongsheng Gao, S.C. Hui, and A. C M Fong. "A multiview facial analysis technique for identity authentication". In: *Pervasive Computing, IEEE* 2.1 (2003), pp. 38–45 (cit. on p. 26).

[78] Yongsheng Gao and M.K.H. Leung. "Face recognition using line edge map". In: *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 24.6 (June 2002), pp. 764 –779 (cit. on p. 25).

[79] Nicolás García-Pedrajas, César García-Osorio, and Colin Fyfe. “Nonlinear Boosting Projections for Ensemble Construction”. In: *Journal of Machine Learning Research* 8 (May 2007), pp. 1–33. URL: http://dl.acm.org/citation.cfm?id=1248659.1248660 (cit. on p. 36).

[80] A.S. Georghiades, P.N. Belhumeur, and D.J. Kriegman. “From Few to Many: Illumination Cone Models for Face Recognition under Variable Lighting and Pose”. In: *IEEE Transactions On Pattern Analysis and Machine intelligence* 23.6 (2001), pp. 643–660 (cit. on pp. 16, 25).

[81] Shaogang Gong, Stephen J. McKenna, and Alexandra Psarrou. *Dynamic Vision: From Images to Face Recognition.* 1st. London, UK, UK: Imperial College Press, 2000 (cit. on p. 27).

[82] Gaile G. Gordon. “Face Recognition from Frontal and Profile Views”. In: *Proceedings of the International Workshop on Automatic Face- and Gesture-Recognition (IWAFGR 95).* 1995, pp. 47–52 (cit. on p. 25).

[83] Arnulf B. A. Graf and Silvio Borer. “Normalization in support vector machines”. In: *DAGM 2001 Pattern Recognition.* Springer, 2001, pp. 277–282 (cit. on p. 32).

[84] A. Hadid et al. “Face and Eye Detection for Person Authentication in Mobile Phones”. In: *Distributed Smart Cameras, 2007. ICDSC ’07. First ACM/IEEE International Conference on.* 2007, pp. 101–108 (cit. on p. 16).

[85] M.D. Hafiz et al. “Towards Identifying Usability and Security Features of Graphical Password in Knowledge Based Authentication Technique”. In: *Modeling Simulation, 2008. AICMS 08. Second Asia International Conference on.* May 2008, pp. 396 –403 (cit. on p. 8).

[86] RichardI. Hartley. “Theory and Practice of Projective Rectification”. English. In: *International Journal of Computer Vision* 35.2 (1999), pp. 115–127. URL: http://dx.doi.org/10.1023/A%3A1008115206617 (cit. on p. 29).

[87] V. Hautamaki et al. “Maximum a Posteriori Adaptation of the Centroid Model for Speaker Verification”. In: *Signal Processing Letters, IEEE* 15 (2008), pp. 162–165 (cit. on p. 10).

[88] Simon Haykin. *Neural Networks: A Comprehensive Foundation.* 2nd. Upper Saddle River, NJ, USA: Prentice Hall PTR, 1998 (cit. on pp. 32, 34).

[89] D. O. Hebb. *The Organization of Behavior.* New York: Wiley, 1949 (cit. on p. 34).

[90] Cormac Herley. "So long, and no thanks for the externalities: the rational rejection of security advice by users". In: *Proceedings of the 2009 workshop on New security paradigms workshop*. NSPW '09. Oxford, United Kingdom: ACM, 2009, pp. 133–144. URL: http://doi.acm.org/10.1145/1719030.1719050 (cit. on p. 2).

[91] Xiaofei He et al. "Face recognition using Laplacianfaces". In: *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 27.3 (2005), pp. 328–340 (cit. on p. 26).

[92] Erik Hjelmås and Boon Kee Low. "Face Detection: A Survey". In: *Computer Vision and Image Understanding* 83.3 (2001), pp. 236–274. URL: http://www.sciencedirect.com/science/article/pii/S107731420190921X (cit. on pp. 17–19, 22).

[93] J. J. Hopfield. "Neurocomputing: foundations of research". In: ed. by James A. Anderson and Edward Rosenfeld. Cambridge, MA, USA: MIT Press, 1988. Chap. Neural networks and physical systems with emergent collective computational abilities, pp. 457–464. URL: http://dl.acm.org/citation.cfm?id=65669.104422 (cit. on p. 34).

[94] C. W. Hsu, C. C. Chang, and C. J. Lin. *A practical guide to support vector classification*. Department of Computer Science and Information Engineering, National Taiwan University. Taipei, Taiwan, 2003. URL: http://www.csie.ntu.edu.tw/\~{}cjlin/papers/guide/guide.pdf (cit. on pp. 62, 68, 83).

[95] Rein-Lien Hsu, Mohamed Abdel-Mottaleb, and Anil K. Jain. "Face Detection In Color Images". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 24.5 (2002), pp. 696–706 (cit. on p. 21).

[96] Di Huang et al. "Local Binary Patterns and Its Application to Facial Image Analysis: A Survey". In: *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews* 41.6 (Nov. 2011), pp. 765–781 (cit. on pp. 22, 27).

[97] C. Iancu, P. Corcoran, and G. Costache. "A Review of Face Recognition Techniques for In-Camera Applications". In: *Signals, Circuits and Systems, 2007. ISSCS 2007. International Symposium on*. Vol. 1. July 2007, pp. 1 –4 (cit. on p. 27).

[98] Y. Ijiri, M. Sakuragi, and Shihong Lao. "Security Management for Mobile Devices by Face Recognition". In: *Mobile Data Management, 2006. MDM 2006. 7th International Conference on*. 2006, pp. 49–49 (cit. on p. 16).

[99] Anil Jain, Lin Hong, and Sharath Pankanti. "Biometric identification". In: *Commun. ACM* 43.2 (Feb. 2000), pp. 90–98. URL: http://doi.acm.org/10.1145/328236.328110 (cit. on p. 9).

[100] Anil K. Jain and Stan Z. Li. *Handbook of Face Recognition*. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2005 (cit. on p. 27).

[101] Oliver Jesorsky, Klaus J. Kirchberg, and Robert Frischholz. "Robust Face Detection Using the Hausdorff Distance". In: *Proceedings of the Third International Conference on Audio- and Video-Based Biometric Person Authentication*. AVBPA '01. London, UK, UK: Springer-Verlag, 2001, pp. 90–95. URL: http://dl.acm.org/citation.cfm?id=646073.677460 (cit. on p. 21).

[102] Michael Jones. "Face Recognition : Where We Are and Where To Go From Here". In: *IEEJ Transactions on Electronics, Information and Systems* 129.5 (May 2009), pp. 770–777. URL: http://ci.nii.ac.jp/naid/10024774945/en/ (cit. on p. 27).

[103] M Kass, A Witkin, and D Terzopoulos. "Snakes - Active Contour Models". English. In: *International Journal Of Computer Vision* 1.4 (1987), pp. 321–331 (cit. on p. 80).

[104] A. Khashman. "Intelligent Local Face Recognition". In: ed. by M. Stewart Bartlett K. Delac M. Grgic. IN-TECH, Vienna, Austria, 2008. Chap. Recent Advances in Face Recognition (cit. on p. 24).

[105] Wolf Kienzle et al. "Face Detection – Efficient and Rank Deficient". In: *Eighteenth Annual Conference on Neural Information Processing Systems (NIPS)*. Feb. 13, 2006. URL: http://dblp.uni-trier.de/db/conf/nips/nips2004.html#KienzleBFS04 (cit. on p. 21).

[106] Dong-Ju Kim, Kwang-Woo Chung, and Kwang-Seok Hong. "Person authentication using face, teeth and voice modalities for mobile device security". In: *Consumer Electronics, IEEE Transactions on* 56.4 (2010), pp. 2678–2685 (cit. on p. 16).

[107] Tomi Kinnunen and Haizhou Li. "An overview of text-independent speaker recognition: From features to supervectors". In: *Speech Communication* 52.1 (2010), pp. 12–40. URL: http://www.sciencedirect.com/science/article/pii/S0167639309001289 (cit. on p. 10).

[108] J. Kittler et al. "3D Assisted Face Recognition: A Survey of 3D Imaging, Modelling and Recognition Approachest". In: *Computer Vision and Pattern Recognition - Workshops, 2005. CVPR Workshops. IEEE Computer Society Conference on*. June 2005, p. 114 (cit. on p. 27).

[109] Brendan Klare and Anil K. Jain. "On a taxonomy of facial features". In: *IEEE International Conference on Biometrics: Theory, Applications, and Systems*. 2010 (cit. on p. 26).

[110] Kurt Konolige. “Small Vision Systems: Hardware and Implementation”. English. In: *Robotics Research*. Ed. by Yoshiaki Shirai and Shigeo Hirose. Springer London, 1998, pp. 203–212. URL: http://dx.doi.org/10.1007/978-1-4471-1580-9_19 (cit. on pp. 29, 66).

[111] Ludmila I. Kuncheva. *Combining Pattern Classifiers: Methods and Algorithms*. Wiley-Interscience, 2004 (cit. on p. 37).

[112] Werasak Kurutach, Rerkchai Fooprateepsiri, and Suronapee Phoomvuthisarn. “A highly robust approach face recognition using hausdorff-trace transformation”. In: *Proceedings of the 17th international conference on Neural information processing: models and applications - Volume Part II*. Sydney, Australia, 2010, pp. 549–556 (cit. on p. 27).

[113] A. Lawson et al. “Survey and evaluation of acoustic features for speaker recognition”. In: *Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on*. 2011, pp. 5444–5447 (cit. on p. 10).

[114] Richard Lengagne, Jean-Philippe Tarel, and Olivier Monga. “From 2D images to 3D face geometry”. In: *Proceedings of the Second International Conference on Automatic Face and Gesture Recognition, 1996*. Oct. 1996, pp. 301–306 (cit. on p. 29).

[115] Rainer Lienhart and Jochen Maydt. “An Extended Set of Haar-Like Features for Rapid Object Detection”. In: *IEEE International Conference on Image Processing 2002*. 2002, pp. 900–903 (cit. on pp. 21, 41, 42, 53, 60, 66, 83).

[116] Shang-Hung Lin, Sun-Yuan Kung, and Long-Ji Lin. “Face recognition/detection by probabilistic decision-based neural network”. In: *Neural Networks, IEEE Transactions on* 8.1 (Jan. 1997), pp. 114–132 (cit. on p. 25).

[117] Ping Li. “ABC-boost: adaptive base class boost for multi-class classification”. In: *Proceedings of the 26th Annual International Conference on Machine Learning*. ICML ’09. Montreal, Quebec, Canada: ACM, 2009, pp. 625–632. URL: http://doi.acm.org/10.1145/1553374.1553455 (cit. on p. 36).

[118] S.Z. Li and Juwei Lu. “Face recognition using the nearest feature line method”. In: *Neural Networks, IEEE Transactions on* 10.2 (Mar. 1999), pp. 439–443 (cit. on p. 25).

[119] Chengjun Liu. “A Bayesian discriminating features method for face detection”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 25.6 (June 2003), pp. 725–740 (cit. on p. 21).

[120] Chengjun Liu and H. Wechsler. “Gabor feature based classification using the enhanced fisher linear discriminant model for face recognition”. In: *Image Processing, IEEE Transactions on* 11.4 (Apr. 2002), pp. 467–476 (cit. on p. 26).

[121] Xuchun Li, Lei Wang, and Eric Sung. “AdaBoost with SVM-based component classifiers”. In: *Engineering Applications of Artificial Intelligence* 21.5 (Aug. 2008), pp. 785–795. URL: http://dx.doi.org/10.1016/j.engappai.2007.07.001 (cit. on p. 36).

[122] Hong Lu et al. “SpeakerSense: energy efficient unobtrusive speaker identification on mobile phones”. In: *Proceedings of the 9th international conference on Pervasive computing*. Pervasive’11. San Francisco, USA: Springer-Verlag, 2011, pp. 188–205. URL: http://dl.acm.org/citation.cfm?id=2021975.2021992 (cit. on p. 10).

[123] H. Manabe et al. “Security Evaluation of Biometrics Authentications for Cellular Phones”. In: *Intelligent Information Hiding and Multimedia Signal Processing, 2009. IIH-MSP ’09. Fifth International Conference on*. Sept. 2009, pp. 34 –39 (cit. on p. 15).

[124] A.M. Martinez and A.C. Kak. “PCA versus LDA”. In: *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 23.2 (Feb. 2001), pp. 228–233 (cit. on p. 19).

[125] Birgitta Martinkauppi. “Face colour under varying illumination - analysis and applications”. PhD thesis. University of Oulu, 2002 (cit. on p. 21).

[126] Iain Matthews and Simon Baker. “Active Appearance Models Revisited”. In: *International Journal of Computer Vision* 60.2 (Nov. 2004), pp. 135–164. URL: http://dx.doi.org/10.1023/B:VISI.0000029666.37597.d3 (cit. on p. 67).

[127] Rene Mayrhofer and Thomas Kaiser. “Towards usable authentication on mobile phones: An evaluation of speaker and face recognition on off-the-shelf handsets”. In: *Proc. IWSSI/SPMU 2012: 4th International Workshop on Security and Privacy in Spontaneous Interaction and Mobile Phone Use, colocated with Pervasive 2012*. Newcastle, UK, June 18, 2012 (cit. on p. 16).

[128] C. McCool et al. “Bi-Modal Person Recognition on a Mobile Phone: Using Mobile Phone Data”. In: *Multimedia and Expo Workshops (ICMEW), 2012 IEEE International Conference on*. 2012, pp. 635–640 (cit. on p. 16).

[129] Warren Mcculloch and Walter Pitts. “A Logical Calculus of Ideas Immanent in Nervous Activity”. In: *Bulletin of Mathematical Biophysics* 5 (1943), pp. 127–147 (cit. on p. 34).

[130] Joo Er Meng et al. “Face recognition with radial basis function (RBF) neural networks”. In: *Neural Networks, IEEE Transactions on* 13.3 (May 2002), pp. 697–710 (cit. on p. 25).

[131] Ji Ming et al. “Robust Speaker Recognition in Noisy Conditions”. In: *Audio, Speech, and Language Processing, IEEE Transactions on* 15.5 (2007), pp. 1711–1723 (cit. on p. 10).

[132] Thomas M. Mitchell. *Machine Learning*. Ed. by Eric Munson. 1st ed. New York, NY, USA: McGraw-Hill, Inc., 1997 (cit. on pp. 25, 60, 66, 68).

[133] M. Muaaz and C. Nickel. “Influence of different walking speeds and surfaces on accelerometer-based biometric gait recognition”. In: *Telecommunications and Signal Processing (TSP), 2012 35th International Conference on*. 2012, pp. 508–512 (cit. on p. 10).

[134] M. P. Murray. “Gait as a total pattern of movement: Including a bibliography on gait”. In: *American Journal of Physical Medicine & Rehabilitation* 46.1 (1967), p. 290 (cit. on p. 10).

[135] Pat M. Murray, Bernard A. Drought, and Ross C. Kory. “Walking Patterns of Normal Men”. In: *The Journal of Bone & Joint Surgery* 46.2 (Mar. 1, 1964), pp. 335–360 (cit. on p. 10).

[136] I. Muslukhov et al. “Understanding Users’ Requirements for Data Protection in Smartphones”. In: *Data Engineering Workshops (ICDEW), 2012 IEEE 28th International Conference on*. 2012, pp. 228–235 (cit. on p. 3).

[137] S.A. Nazeer, N. Omar, and M. Khalid. “Face Recognition System using Artificial Neural Networks Approach”. In: *Signal Processing, Communications and Networking, 2007. ICSCN ’07. International Conference on*. Feb. 2007, pp. 420–425 (cit. on p. 26).

[138] T. Ojala, M. Pietikainen, and T. Maenpaa. “Multiresolution gray-scale and rotation invariant texture classification with local binary patterns”. In: *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 24.7 (2002), pp. 971–987 (cit. on p. 26).

[139] Gang Pan, Zhaohui Wu, and Lin Sun. “Liveness Detection for Face Recognition”. In: ed. by Kresimir Delac, Mislav Grgic, and Marian Stewart Bartlett. InTech, 2008. Chap. Recent Advances in Face Recognition, p. 236. URL: http://www.intechopen.com/books/recent_advances_in_face_recognition/liveness_detection_for_face_recognition (cit. on p. 14).

[140] C.P. Papageorgiou, M. Oren, and T. Poggio. “A general framework for object detection”. In: *Computer Vision, 1998. Sixth International Conference on*. 1998, pp. 555–562 (cit. on p. 21).

[141] Shwetak N. Patel, Jeffrey S. Pierce, and Gregory D. Abowd. “A gesture-based authentication scheme for untrusted public terminals”. In: *Proceedings of the 17th annual ACM symposium on User interface software and technology*. UIST ’04. Santa Fe, NM, USA: ACM, 2004, pp. 157–160. URL: http://doi.acm.org/10.1145/1029632.1029658 (cit. on p. 9).

[142] A. Pentland, B. Moghaddam, and T. Starner. “View-based and modular eigenspaces for face recognition”. In: *IEEE Computer Society Conference on Computer Vision and Pattern Recognition 1994 (CVPR ’94)*. June 1994, pp. 84–91 (cit. on p. 55).

[143] P. Jonathon Phillips. “Support Vector Machines Applied to Face Recognition”. In: *Neural Information Processing Systems*. Ed. by M. I. Jordan, M. J. Kearns, and S. A. Solla. Vol. 10. 1998, pp. 803–809 (cit. on pp. 60, 66, 68).

[144] P.J. Phillips et al. “Distinguishing identical twins by face recognition”. In: *Automatic Face Gesture Recognition and Workshops (FG 2011), 2011 IEEE International Conference on*. 2011, pp. 185–192 (cit. on p. 90).

[145] P.J. Phillips et al. “Overview of the face recognition grand challenge”. In: *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*. Vol. 1. June 2005, pp. 947–954 (cit. on p. 16).

[146] J. R. Quinlan. “Bagging, Boosting, and C4.5”. In: *In Proceedings of the Thirteenth National Conference on Artificial Intelligence*. AAAI Press, 1996, pp. 725–730 (cit. on p. 37).

[147] Ashwini Rao, Birendra Jha, and Gananand Kini. “Effect of grammar on security of long passwords”. In: *Proceedings of the third ACM conference on Data and application security and privacy*. CODASPY ’13. San Antonio, Texas, USA: ACM, 2013, pp. 317–324. URL: http://doi.acm.org/10.1145/2435349.2435395 (cit. on p. 2).

[148] K.S. Rao et al. “Robust speaker recognition on mobile devices”. In: *Signal Processing and Communications (SPCOM), 2010 International Conference on*. 2010, pp. 1–5 (cit. on p. 10).

[149] Z. Riaz, A. Gilgiti, and S.M. Mirza. “Face recognition: a review and comparison of HMM, PCA, ICA and neural networks”. In: *E-Tech 2004*. July 2004, pp. 41 –46 (cit. on p. 26).

[150] Saharon Rosset. “Robust boosting and its relation to bagging”. In: *Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*. KDD ’05. Chicago, Illinois, USA: ACM, 2005, pp. 249–255. URL: http://doi.acm.org/10.1145/1081870.1081900 (cit. on p. 36).

[151] Carsten Rother, Vladimir Kolmogorov, and Andrew Blake. “"GrabCut": interactive foreground extraction using iterated graph cuts”. In: *ACM Transactions on Graphics* 23.3 (Aug. 2004), pp. 309–314. URL: http://doi.acm.org/10.1145/1015706.1015720 (cit. on p. 67).

[152] H. A. Rowley, S. Baluja, and T. Kanade. “Rotation Invariant Neural Network-Based Face Detection”. In: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. CVPR ’98. Washington, DC, USA: IEEE Computer Society, 1998, pp. 38–. URL: http://dl.acm.org/citation.cfm?id=794191.794781 (cit. on p. 21).

[153] Henry A. Rowley, Shumeet Baluja, and Takeo Kanade. “Neural Network-Based Face Detection”. In: *IEEE Transactions On Pattern Analysis and Machine intelligence* 20.1 (1998), pp. 23–38 (cit. on p. 21).

[154] William Rucklidge. *Efficient Visual Recognition Using the Hausdorff Distance*. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 1996 (cit. on pp. 21, 27).

[155] Hossein Sahoolizadeh, Davood Sarikhanimoghadam, and Hamid Dehghani. “Face Detection using Gabor Wavelets and Neural Networks”. In: *World Academy of Science, Engineering and Technology* 21.97 (Sept. 2008), pp. 552–554 (cit. on p. 21).

[156] Joaquim Salvi, Jordi Pagès, and Joan Batlle. “Pattern codification strategies in structured light systems”. In: *Pattern Recognition* 37.4 (2004), pp. 827–849. URL: http://www.sciencedirect.com/science/article/pii/S0031320303003303 (cit. on p. 28).

[157] Modesto Castrillón Santana et al. “A comparison of face and facial feature detectors based on the Viola-Jones general object detection framework”. In: *Machine Vision and Applications* 22.3 (2011), pp. 481–494. URL: http://dblp.uni-trier.de/db/journals/mva/mva22.html#SantanaDHL11 (cit. on pp. 22, 61).

[158] Modesto Castrillón Santana et al. “Face and Facial Feature Detection Evaluation - Performance Evaluation of Public Domain Haar Detectors for Face and Facial Feature Detection”. In: *International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications*. Ed. by Alpesh Ranchordas and Helder Araújo. Vol. 2. Apr. 7, 2008, pp. 167–172 (cit. on pp. 58, 59, 61, 66).

[159] Robert E. Schapire and Yoram Singer. “Improved Boosting Algorithms Using Confidence-rated Predictions”. In: *Machine Learning* 37.3 (Dec. 1999), pp. 297–336. URL: http://dx.doi.org/10.1023/A:1007614523901 (cit. on p. 36).

[160] D. Scharstein and R. Szeliski. "High-accuracy stereo depth maps using structured light". In: *Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on.* Vol. 1. 2003, pp. 195–202 (cit. on p. 28).

[161] Florian Schaub, Ruben Deyhle, and Michael Weber. "Password entry usability and shoulder surfing susceptibility on different smartphone platforms". In: *Proceedings of the 11th International Conference on Mobile and Ubiquitous Multimedia.* MUM '12. Ulm, Germany: ACM, 2012, 13:1–13:10. URL: http://doi.acm.org/10.1145/2406367.2406384 (cit. on p. 7).

[162] Alize Scheenstra, Arnout Ruifrok, and Remco C. Veltkamp. "A survey of 3d face recognition methods". In: *Proceedings of the 5th international conference on Audio- and Video-Based Biometric Person Authentication.* AVBPA'05. Hilton Rye Town, NY: Springer-Verlag, 2005, pp. 891–899. URL: http://dx.doi.org/10.1007/11527923_93 (cit. on p. 27).

[163] Roland Schlöglhofer and Johannes Sametinger. "Secure and usable authentication on mobile devices". In: *Proceedings of the 10th International Conference on Advances in Mobile Computing and Multimedia.* MoMM '12. Bali, Indonesia: ACM, 2012, pp. 257–262. URL: http://doi.acm.org/10.1145/2428955.2429004 (cit. on p. 9).

[164] Henry Schneiderman and Takeo Kanade. "Object Detection Using the Statistics of Parts". In: *International Journal of Computer Vision* 56.3 (Feb. 2004), pp. 151–177 (cit. on p. 21).

[165] H. Schneiderman and T. Kanade. "A statistical method for 3D object detection applied to faces and cars". In: *Computer Vision and Pattern Recognition, 2000. Proceedings. IEEE Conference on.* Vol. 1. IEEE Computer Soc, June 2000, pp. 746–751 (cit. on p. 21).

[166] H. Schneiderman and T. Kanade. "Probabilistic Modeling of Local Appearance and Spatial Relationships for Object Recognition". In: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition.* CVPR '98. Washington, DC, USA: IEEE Computer Society, 1998, pp. 45–. URL: http://dl.acm.org/citation.cfm?id=794191.794783 (cit. on p. 21).

[167] M.P. Segundo et al. "Automatic Face Segmentation and Facial Landmark Detection in Range Images". In: *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on* 40.5 (Oct. 2010), pp. 1319–1330 (cit. on p. 67).

[168] Helmut Seibert. “Efficient Segmentation of 3D Face Reconstructions”. In: *Intelligent Information Hiding and Multimedia Signal Processing (IIH-MSP), 2012 Eighth International Conference on*. July 2012, pp. 31–34 (cit. on p. 67).

[169] Julian Seifert et al. “TreasurePhone: Context-Sensitive User Data Protection on Mobile Phones”. In: *Pervasive 2010*. Vol. Volume 6030/2010. Lecture Notes in Computer Science. Helsinki, Finland: Springer Berlin Heidelberg, 2010, pp. 130–137. URL: http://dx.doi.org/10.1007/978-3-642-12654-3_8 (cit. on p. 9).

[170] Linlin Shen et al. “Secure mobile services by face and speech based personal authentication”. In: *Intelligent Computing and Intelligent Systems (ICIS), 2010 IEEE International Conference on*. Vol. 3. 2010, pp. 97–100 (cit. on p. 16).

[171] L. Sirovich and M. Kirby. “Low-Dimensional Procedure for the Characterization of Human Faces”. In: *Journal of the Optical Society of America A* 4.3 (1987), pp. 519–524 (cit. on pp. 24, 29).

[172] Marina Skurichina and Robert P. W. Duin. “Bagging, Boosting and the Random Subspace Method for Linear Classifiers”. English. In: *Pattern Analysis and Applications* 5.2 (2002), pp. 121–135. URL: http://dx.doi.org/10.1007/s100440200011 (cit. on p. 37).

[173] Marina Skurichina and Robert P.W. Duin. “Bagging for linear classifiers”. In: *Pattern Recognition* 31.7 (1998), pp. 909–930. URL: http://www.sciencedirect.com/science/article/pii/S0031320397001106 (cit. on p. 37).

[174] Kah-kay Sung and Tomaso Poggio. “Example-based learning for view-based human face detection”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 20 (1998), pp. 39–51 (cit. on p. 21).

[175] Qian Tao and R. Veldhuis. “Biometric Authentication System on Mobile Personal Devices”. In: *Instrumentation and Measurement, IEEE Transactions on* 59.4 (2010), pp. 763–773 (cit. on p. 15).

[176] Furkan Tari, A. Ant Ozok, and Stephen H. Holden. “A comparison of perceived and real shoulder-surfing risks between alphanumeric and graphical passwords”. In: *Proceedings of the second symposium on Usable privacy and security*. SOUPS ’06. Pittsburgh, Pennsylvania: ACM, 2006, pp. 56–66. URL: http://doi.acm.org/10.1145/1143120.1143128 (cit. on p. 7).

[177] M.H. Teja. “Real-time live face detection using face template matching and DCT energy analysis”. In: *Soft Computing and Pattern Recognition (SoCPaR), 2011 International Conference of*. Oct. 2011, pp. 342 –346 (cit. on p. 14).

[178] P. Tresadern et al. “Mobile Biometrics: Combined Face and Voice Verification for a Mobile Platform”. In: *IEEE Pervasive Computing* 12.1 (2013), pp. 79–87 (cit. on p. 16).

[179] R. Tronci et al. “Fusion of multiple clues for photo-attack detection in face recognition systems”. In: *International Joint Conference on Biometrics*. Oct. 2011, pp. 1–6 (cit. on p. 14).

[180] F Tsalakanidou, D Tzovaras, and M.G Strintzis. “Use of depth and colour eigenfaces for face recognition”. In: *Pattern Recognition Letters* 24.9–10 (2003), pp. 1427–1435. URL: http://www.sciencedirect.com/science/article/pii/S0167865502003835 (cit. on p. 26).

[181] Matthew Turk and Alex Pentland. “Eigenfaces for recognition”. In: *Cognitive Neuroscience* 3.1 (Jan. 1991), pp. 71–86 (cit. on pp. 20, 24, 29, 31, 53, 56, 62).

[182] V. Vapnik and A. Lerner. “Pattern Recognition using Generalized Portrait Method”. In: *Automation and Remote Control* 24.6 (1963), pp. 774–780 (cit. on p. 32).

[183] Christopher L. Vaughan, Brian L. Davis, and Jeremy C. O’Connor. *Dynamics of Human Gait*. Ed. by Christopher L. Vaughan. Second. Howard Place, Western Cape 7450, South Africa: Kiboho Publishers, 1999. URL: http://www.kiboho.co.za/GaitCD/GaitBook.pdf (cit. on pp. 10, 11).

[184] Krithika Venkataramani, S. Qidwai, and B. Vijayakumar. “Face authentication from cell phone camera images with illumination and temporal variations”. In: *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on* 35.3 (2005), pp. 411–418 (cit. on p. 26).

[185] Paul Viola and Michael Jones. “Robust real-time face detection”. In: *International Journal of Computer Vision* 57 (2004), pp. 137–154 (cit. on p. 21).

[186] P. Viola and M. Jones. “Rapid object detection using a boosted cascade of simple features”. In: *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition* 1 (2001), pp. 511–518 (cit. on pp. 21, 41, 42, 53, 60, 66, 83).

[187] Michael Wagner. “Liveness Assurance in Voice Authentication”. In: *Encyclopedia of Biometrics*. Ed. by StanZ. Li and Anil Jain. Springer US, 2009, pp. 916–924. URL: http://dx.doi.org/10.1007/978-0-387-73003-5_70 (cit. on p. 10).

[188] Michael Wagner and Girija Chetty. “Liveness Assurance in Face Authentication”. In: *Encyclopedia of Biometrics*. Ed. by Stan Z. Li and Anil Jain. Springer US, 2009, pp. 908–916. URL: http://dx.doi.org/10.1007/978-0-387-73003-5_67 (cit. on p. 14).

[189] Harry Wechsler. *Reliable Face Recognition Methods: System Design, Implementation and Evaluation (International Series on Biometrics)*. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2006 (cit. on p. 27).

[190] Fabian Wenny. "Stereo Vision and 3D Face Segmentation Techniques for Mobile Devices". *working title, to be published.* MA thesis. Softwarepark 11, 4232 Hagenberg/Austria: Department of Mobile Computing, School of Informatics, Communication and Media, University of Applied Sciences Upper Austria, Sept. 2013 (cit. on p. 40).

[191] B. Weyrauch et al. "Component-Based Face Recognition with 3D Morphable Models". In: *Conference on Computer Vision and Pattern Recognition Workshop, 2004. (CVPRW '04)*. June 2004, p. 85 (cit. on p. 26).

[192] Michael Whittle. *Gait analysis: an introduction*. 3rd ed. Elsevier, 2002 (cit. on p. 10).

[193] Susan Wiedenbeck et al. "Design and evaluation of a shoulder-surfing resistant graphical password scheme". In: *Proceedings of the working conference on Advanced visual interfaces*. AVI '06. Venezia, Italy: ACM, 2006, pp. 177–184. URL: http://doi.acm.org/10.1145/1133265.1133303 (cit. on p. 8).

[194] Laurenz Wiskott et al. "Face Recognition By Elastic Bunch Graph Matching". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 19 (1997), pp. 775–779 (cit. on p. 25).

[195] Ian H. Witten, Eibe Frank, and Mark A. Hall. *Data Mining: Practical Machine Learning Tools and Techniques*. Ed. by Jim Gray. 3. Amsterdam: Morgan Kaufmann, 2011. URL: http://www.sciencedirect.com/science/book/9780123748560 (cit. on p. 31).

[196] Chenyang Xu and J.L. Prince. "Snakes, shapes, and gradient vector flow". In: *Image Processing, IEEE Transactions on* 7.3 (1998), pp. 359–369 (cit. on pp. 42, 80).

[197] Chenyang Xu, Jr. Yezzi A., and J.L. Prince. "On the relationship between parametric and geometric active contours". In: *Conference on Signals, Systems and Computers, 2000. Conference Record of the Thirty-Fourth Asilomar*. Vol. 1. 2000, 483–489 vol.1 (cit. on p. 80).

[198] Ming-Hsuan Yang, D.J. Kriegman, and N. Ahuja. "Detecting faces in images: a survey". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 24.1 (Jan. 2002), pp. 34–58 (cit. on p. 22).

[199] Benjamin D. Zarit, Boaz J. Super, and Francis K. H. Quek. "Comparison of Five Color Models in Skin Pixel Classification". In: *International Workshop on Recognition, Analysis, and Tracking of Faces and Gestures in Real-Time Systems*. 1999, pp. 58–63 (cit. on p. 21).

[200] Emanuel Zezschwitz, Alexander Luca, and Heinric Hussmann. "Survival of the Shortest: A Retrospective Analysis of Influencing Factors on Password Composition". In: *Human-Computer Interaction (INTERACT 2013)*. Ed. by Paula Kotzé et al. Vol. 8119. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2013, pp. 460–467. URL: http://dx.doi.org/10.1007/978-3-642-40477-1_28 (cit. on p. 2).

[201] Emanuel von Zezschwitz, Paul Dunphy, and Alexander De Luca. "Patterns in the wild: a field study of the usability of pattern and pin-based authentication on mobile devices". In: *Proceedings of the 15th international conference on Human-computer interaction with mobile devices and services*. MobileHCI '13. Munich, Germany: ACM, 2013, pp. 261–270. URL: http://doi.acm.org/10.1145/2493190.2493231 (cit. on p. 3).

[202] Emanuel von Zezschwitz et al. "Making graphic-based authentication secure against smudge attacks". In: *Proceedings of the 2013 international conference on Intelligent user interfaces*. Santa Monica, California, USA: ACM, 2013, pp. 277–286. URL: http://doi.acm.org/10.1145/2449396.2449432 (cit. on p. 8).

[203] Li Zhang, B. Curless, and S.M. Seitz. "Rapid shape acquisition using color structured light and multi-pass dynamic programming". In: *3D Data Processing Visualization and Transmission, 2002. Proceedings. First International Symposium on*. 2002, pp. 24–36 (cit. on p. 28).

[204] Xiaozheng Zhang and Yongsheng Gao. "Face recognition across pose: A review". In: *Pattern Recognition* 42.11 (Nov. 2009), pp. 2876–2896 (cit. on p. 27).

[205] W. Zhao et al. "Face recognition: A literature survey". In: *ACM Computing Surveys* 35.4 (Dec. 2003), pp. 399–458. URL: http://doi.acm.org/10.1145/954339.954342 (cit. on p. 27).

[206] Xuan Zou, J. Kittler, and K. Messer. "Illumination Invariant Face Recognition: A Survey". In: *First IEEE International Conference on Biometrics: Theory, Applications, and Systems (BTAS 2007)*. Sept. 2007, pp. 1–8 (cit. on p. 27).

[207] Moshe Zviran and William J. Haga. "Password security: an empirical study". In: *Journal of Management Information Systems* 15.4 (Mar. 1999), pp. 161–185. URL: http://dl.acm.org/citation.cfm?id=1189462.1189470 (cit. on p. 2).