Received 21 May 2013
Revised 21 May 2013
Accepted 5 July 2013

Towards pan shot face unlock

Using biometric face information from different perspectives to unlock mobile devices

Rainhard Dieter Findling and Rene Mayrhofer
Department of Mobile Computing,
University of Applied Sciences Upper Austria,
Hagenberg, Austria

Abstract

Purpose – Personal mobile devices currently have access to a significant portion of their user's private sensitive data and are increasingly used for processing mobile payments. Consequently, securing access to these mobile devices is a requirement for securing access to the sensitive data and potentially costly services. The authors propose and evaluate a first version of a pan shot face unlock method: a mobile device unlock mechanism using all information available from a 180° pan shot of the device around the user's head – utilizing biometric face information as well as sensor data of built-in sensors of the device. The paper aims to discuss these issues.

Design/methodology/approach – This approach uses grayscale 2D images, on which the authors perform frontal and profile face detection. For face recognition, the authors evaluate different support vector machines and neural networks. To reproducibly evaluate this pan shot face unlock toolchain, the authors assembled the 2013 Hagenberg stereo vision pan shot face database, which the authors describe in detail in this article.

Findings – Current results indicate that the approach to face recognition is sufficient for further usage in this research. However, face detection is still error prone for the mobile use case, which consequently decreases the face recognition performance as well.

Originality/value – The contributions of this paper include: introducing pan shot face unlock as an approach to increase security and usability during mobile device authentication; introducing the 2013 Hagenberg stereo vision pan shot face database; evaluating this current pan shot face unlock toolchain using the newly created face database.

Keywords Biometric authentication, Face database, Face detection, Face recognition, Face unlock, Mobile device authentication

Paper type Research paper

1. Introduction

Personal mobile devices such as smartphones hold an increasing amount of private and sensitive user data and have access to payment and banking services. Secure access to these personal devices is a prerequisite for securing the stored sensitive data.

A preliminary version of this work was published in MoMM 2012 with a limited set of classifiers and a significantly smaller data set used for evaluation. This work has been carried out by u'smile, the Josef Ressel Centre for User-Friendly Secure Mobile Environments. The authors 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. The authors further want to thank Christoper Gabler for assisting with the creation of the 2013 Hagenberg stereo vision pan shot face database.

International Journal of Pervasive Computing and Communications
Vol. 9 No. 3, 2013
pp. 190-208
© Emerald Group Publishing Limited
1742-7371
DOI 10.1108/IJPCC-05-2013-0012

Therefore, all major mobile device platforms implement a device lock mechanism, which:

- locks the device after a certain time of inactivity; and
- provides different methods to unlock the device immediately before using it.

The most common device unlocking mechanisms are: slide-to-unlock with no security measures whatsoever, entering a PIN or password, and drawing an unlock pattern on the touch screen of the device.

These unlocking mechanisms are prone to the shoulder surfing attack (Schaub *et al.*, 2012; Tari *et al.*, 2006), and other attacks, such as the smudge attack (Aviv *et al.*, 2010; von Zezschwitz *et al.*, 2013). With the shoulder surfing attack, an attacker watches (directly or indirectly, e.g. via video cameras) the display while the legitimate user authenticates, and thereby observes the shared secret. With the smudge attack, an attacker analyzes the display of the mobile device after the legitimate user successfully authenticated and thereby observes the pattern that remains on the display after drawing the unlock pattern due to residual grease left by unclothed fingers. As the shared secret is revealed with both attacks, the attacker can authenticate with the device just as the legitimate user can.

Face unlock is an approach to unlocking a mobile device based on biometric information of the user's face. It aims to:

- be more robust against attacks such as the shoulder surfing and smudge attack; and
- increase the usability of unlocking, as it is potentially faster than entering a PIN or password – and, possibly even more importantly, without the user needing to remember a shared secret.

Unfortunately, using only frontal face information for face-based authentication can still be circumvented by a photo attack (Anjos and Marcel, 2011; Bao *et al.*, 2009; Frischholz and Werner, 2003; Tronci *et al.*, 2011; Wagner and Chetty, 2009): an attacker performs a spoof authentication by presenting a sufficiently large and high-quality photograph or video of the authorized person to the camera. For many people, the required data can be grabbed from social networks or video platforms without restrictions and at no costs.

Therefore, further extensions to mobile user authentication are required – such as using more data or combining different authentication methods. Recent research (Tresadern *et al.*, 2013; Mayrhofer and Kaiser, 2012) combines real time face and voice recognition in order to improve security when authenticating at a mobile device.

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 the user's face, see Figure 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 attack, 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).

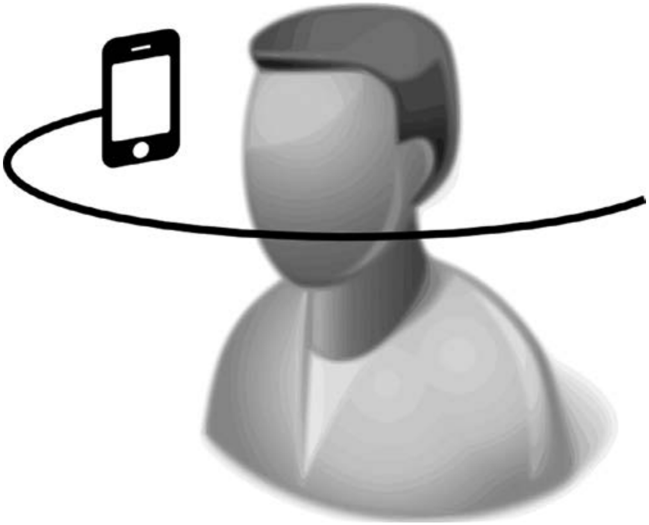

Figure 1.
The mobile device records the user's face during a pan shot

In this article, we make the following novel contributions:

- We introduce pan shot face unlock, which is an approach to increase security and usability of unlocking a mobile device, using biometric face information from a 180° pan shot around the users head (Section 3).
- We introduce the 2013 Hagenberg stereo vision pan shot face database, which was assembled for evaluating pan shot face unlock approaches. The database encompasses 30 people with 20 pan shots for each person and recording device (2D grayscale image, stereo camera image pair, and range image). Each pan shot consists of nine images for a total angle of 180° (Section 4).
- We evaluate the capabilities of our pan shot face unlock toolchain: first, we perform face detection on pan shot images and analyze the detection rate. Based on detected faces, we then perform face recognition using different classifiers and analyze the recognition rate and combine the classifier outputs for analyzing the toolchain's overall recognition capabilities (Section 5).

We note that, as 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). Our approach therefore does not target high security systems at the time of this writing, but is intended to be used for personal devices that are in frequent use – where this approach is more convenient to use and still provides a higher security level than current approaches. Improving the security level of this approach seems possible using techniques such as fuzzy cryptography to alleviate the problem of stored biometric templates, but is currently out of scope of our work.

2. Related work

Face detection

Many different approaches to face detection have been proposed, implemented and evaluated in past literature. For a more comprehensive review we refer to Hjelmås and Low (2001), Huang *et al.* (2011), Santana *et al.* (2011) and Yang *et al.* (2002). Besides the *de-facto* standard approach of Viola and Jones (2001) with Lienhart and Maydt (2002), there exist many other, very diverse approaches.

Turk and Pentland (1991) initially proposed the use of principal component analysis (PCA) and Eigenface subspace for face detection as well as recognition. Sung and Poggio (1998) used view-based model clusters that distinguish between "face" and "non-face". Rowley *et al.* (1998) use neural networks to estimate if parts of an image contain a face. Schneiderman and Kanade (2004) identify objects – including faces – by using wavelet transformation. Sahoolizadeh *et al.* (2008) combine Gabor wavelets and neural networks for face detection and recognition. Bayesian discriminating features were used by Liu (2003), which compare likelihood density estimations of an image to decide if an image contains a face. Finally, the use of skin color for face detection was investigated by different authors, e.g. Hsu *et al.* (2002), Martinkauppi (2002) and Zarit *et al.* (1999), but turned out to be less reliable than other approaches.

Viola and Jones (2001) proposed cascades of boosted classifiers, which are now considered one of the standard approaches to face detection. With the extension by Lienhart and Maydt (2002), this approach to face detection is used by default, e.g. in OpenCV. The boosted classifiers are trained for effective, haar-like features from positive and negative face examples, which have to be provided during a training phase. Those features represent oriented contrast in an image to identify faces in the next step. The classifiers constructed are "weak" classifiers as the basis for standard boosting to obtain "strong" classifiers. Those strong classifiers are then combined to form a classifier cascade. The faster classifiers of the cascade are applied first in the processing chain: as soon as an image fails to pass one classifier, it will not be presented to subsequent classifiers. Finally, to find faces of different sizes, a sliding window principle is applied.

We are currently using the implementation of this algorithm in OpenCV for our approach. There exist different improvements based on the approach of Viola and Jones, such as the approach of Abiantun and Savvides (2009), which boosts the classifiers results using dynamic three-bin real AdaBoost. Although this standard approach works sufficiently well for our current experiments, improvements to face detection especially for non-frontal face images are subject to future work to improve the overall recognition rate (but out of scope of this article).

Face recognition

Again, many different approaches to face recognition have been published in the past 20 years, and for a more comprehensive review we refer to Abate *et al.* (2007), Bowyer *et al.* (2006), Jain and Li (2005), Wechsler (2006), Zhang and Gao (2009), Zhang *et al.* (2003) and Zou *et al.* (2007).

One of the most important approaches was using Eigenfaces for face detection and recognition by Turk and Pentland (1991), which is still used as a baseline for benchmarking newer approaches to face recognition. As an established standard (baseline) approach, we used Eigenfaces for face recognition for preliminary experiments in our Android prototype. Eigenfaces are more prone to recognition errors caused by changes in illumination than Fisherfaces, as shown by Belhumeur *et al.* (1997). Other approaches than dimensionality reduction were also proposed, like Deformable Templates by Yuille *et al.* (1992) or Elastic Bunch Graph Mapping by Wiskott *et al.* (1997).

However, many of the existing face detection and recognition mechanisms were designed to work for frontal face images, without considering profile face images in the first place. One approach to use those mechanisms for more than frontal face images was mentioned by Pentland *et al.* (1994), who used different models for different points of view. Our face recognition approach was inspired by this use of multiple models for different perspectives, but we extend it to utilize support vector machines (SVM) as proposed by Phillips (1998), and neural networks, as mentioned by Mitchell (1997). Comparing results with different classifiers specifically with the inclusion of non-frontal face images is one of the main points in this article.

3. Method

Intended pan shot face unlock usage

The face unlock we are working on 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 quick, non-standardized swipe of the user's mobile phone around the front side of their head, in our current state of prototypical implementation we require the user to perform a more formalized swipe of the camera: the user holds their mobile phone either right or left of their 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 their 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 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 (potentially improved by a trivially simulated eye blinking to overcome the newest checks in current 2D face unlock implementations).

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:

- train classifiers; and
- 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 2 shows an overview of this toolchain, as it has been implemented for the Android prototype and as simulated for improvements on desktop computers.

Recording data and performing 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 their frontal face sections towards the other ear. As soon as the frontal camera points to the other ear – as determined by the gyroscope data – the application goes from ACTIVE to IDLE again. In our current implementation, changing STATE is done by the user pressing a button, while we assume future implementations to trigger this state change automatically based on gyroscope and/or image data.

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 as determined 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 (when using images from our face database for quantitative evaluation, α is 22.5° as used during database image recording to minimize the time required for each of the subjects). Each photo is stored along with meta data (most importantly the current device angle). Therefore, roughly the same number of photos is taken for a pan shot done for each face unlock, and processing the photos can be done afterwards.

We do not currently record a full video stream of the whole camera movement across the user's face because of 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 processing the full video stream for face recognition in real time. We expect the next generation of Android phones to be capable of such processing by utilizing programmable GPUs, and therefore see the possibility to extend our approach to full video streams in future work. However, the basic toolchain can remain unchanged.

As the application switches from ACTIVE to IDLE, the following steps are processed: first, a normalization of the meta data 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 2β is the total angle the mobile phone has rotated, the normalization is performed so that the maximum left angle of all photos is roughly $-\beta$, and the maximum right angle of all photos is roughly β. 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, we perform face detection to each photo. We use the approach of Viola and Jones (2001) with the extension by Lienhart and Maydt (2002), implemented

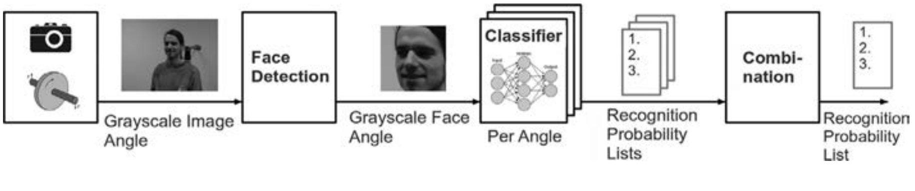

Figure 2.
Overview of the modules used in the pan shot face unlock toolchain

in OpenCV. The detection classifier cascade is chosen depending on the meta data stored along with each photo, where γ is the device angle the photo was shot at and φ is a predefined threshold angle. If $\gamma < -\varphi$, the PROFILE classifier cascade is chosen. If $\gamma > \varphi$, the picture is mirrored – as the OpenCV PROFILE classifier cascade only detects left profile faces – and the PROFILE classifier cascade is chosen. If $|\gamma| \leq \varphi$, 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. Plate 1 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 currently 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.* (1994). Therefore, face images shot at similar angles will be assigned the same or a neighboring classifier in the normal case. For our experiments, we used $\alpha = 22.5°$, which results in about nine 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 detected face images as input, as described in section "Recording data and performing face detection". 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. Depending on the angle of recording, detected face images are assigned to a classifier, which stores the image. This is required to perform a retraining at a later point in time; otherwise, no future expansion with additional image data will be possible. As the user switches the face unlock application from TRAIN to CLASSIFY, each classifier is trained with all face images currently assigned to it. This training is done on the end-user phone without requiring server-assisted ("cloud") computation for privacy reasons. Our current 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 estimated from these distance lists.

There are different ways of combining these probabilities for each person to compute a single probability describing the likelihood that this person was shown in the pan shot pictures: first, adding all probabilities from the different classifiers: a single bad classification result will not cause the overall probability to decrease significantly, making this option more robust. For successfully unlocking the device, a pan shot with many pictures and an overall good probability per detected face is required – otherwise, an attacker could present fewer pictures with known high probability values only and thwart the system.

The second possibility is multiplying all probabilities from the different classifiers: a single bad classification result will cause the overall probability to be dominated

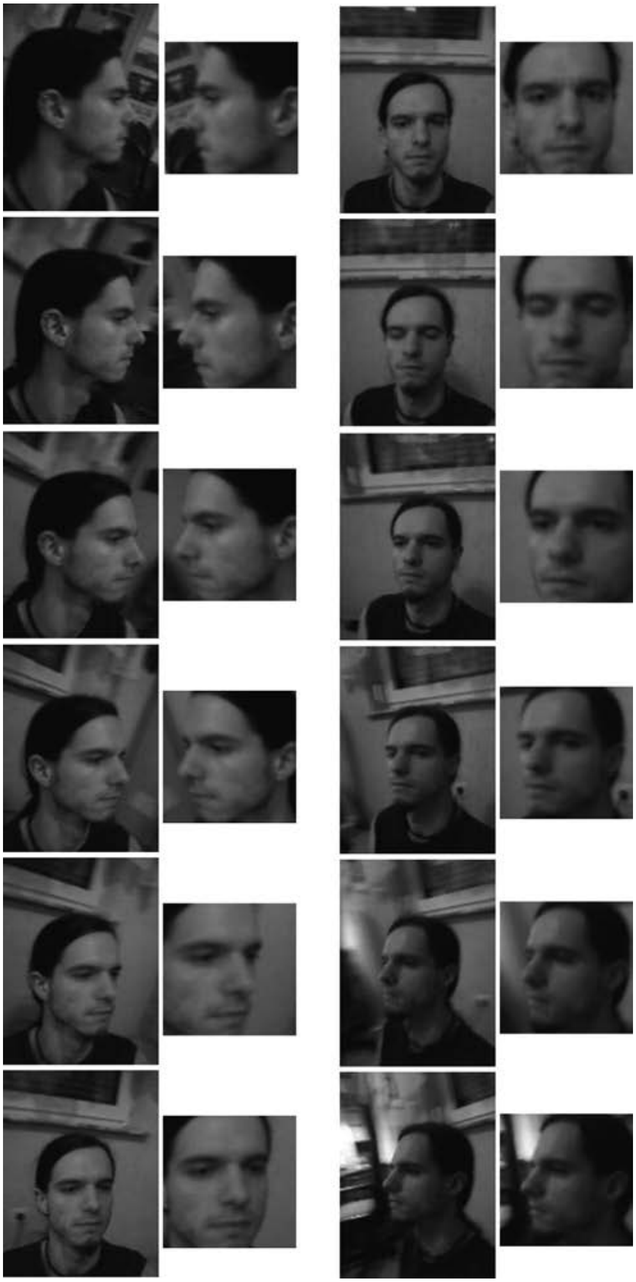

Plate 1.
Pictures recorded
and faces detected
from one pan shot

by the smallest one, and therefore probably result in overall authentication failure. An attacker could artificially increase the total probability by presenting fewer images with known high probability values only, as the total probability will decrease when more pictures are used.

Another possibility is to compute the weighted sum of all probabilities from the different classifiers: while this is essentially the same as adding all the probabilities when assuming the same amount of faces found for pan shots, an attacker could again artificially increase the total probability by presenting fewer images with known, high probability values only. The advantage of a weighted sum is the possibility to parameterize the false positives and false negatives rates depending on the achieved classification accuracy for the different face angles. Based on current, anecdotal experience, recognition based on frontal images is expected to be more accurate than recognition based on profile images. It therefore seems potentially advantageous to assign higher weights to frontal images during the combination of probability values. However, quantitative analysis and optimization of this weight vector is still subject to future work.

The last possibility is computing the weighted product of all probabilities from the different classifiers: again, presenting fewer images with a higher probability could possibly increase the total probability compared to more images containing at least one medium to low probability. Compared to a weighted sum, this option is expected to be more secure (but less robust).

Environment

As environment for a face unlock prototype we are targeting a 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 our prototype implementation, we used a Google/Samsung Nexus S GT-I9023 device, currently running Android 2.3.3. As proof-of-concept and baseline for comparison, we use the Eigenface approach of Turk and Pentland (1991) for face recognition. We further compute the sum of all obtained recognition probabilities from classifiers of different perspectives to obtain an overall probability, with which access to the mobile phone can then either be granted or denied. In preliminary experiments (Findling and Mayrhofer, 2012), we obtained an overall person recognition rate of 55.8 percent – and identified Eigenfaces for face recognition as main component insufficient for further usage in our toolchain.

To evaluate the benefit of improving specific parts of the toolchain used in the prototype, we currently 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 the 2013 Hagenberg stereo vision pan shot face database as data source.

4. The 2013 Hagenberg stereo vision pan shot face database

For doing comprehensive tests on our face recognition approach, we have created the 2013 Hagenberg stereo vision pan shot face database. This database is designed to 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. 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 nine 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 (Figure 3).

For each person, pan shot image set, and perspective, the following images are contained in the database (Figure 4):

- A high quality, colored, 2D image, recorded with a digital single-lens reflex camera (model: Canon EOS 400D, image resolution: 3,888 px × 2,592 px).
- A colored, 2D image pair, recorded with a mobile device stereo camera (model: LG Optimus 3D Max P720, image resolution: 2 × 640 px × 480 px).
- A colored 2D and a raw range image, recorded with a Microsoft Kinect and the OpenKinect framework (image resolution: 2 × 640 px × 480 px).

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 I 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 Plate 2.

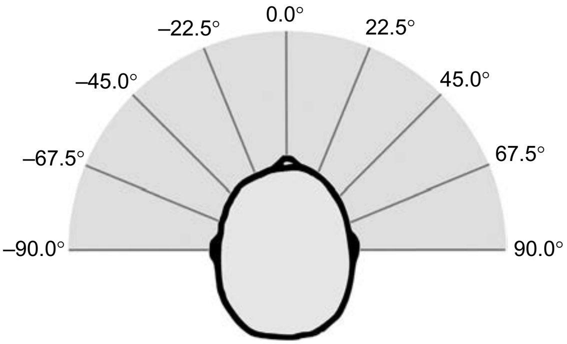

Figure 3.
Angles at which images have been recorded

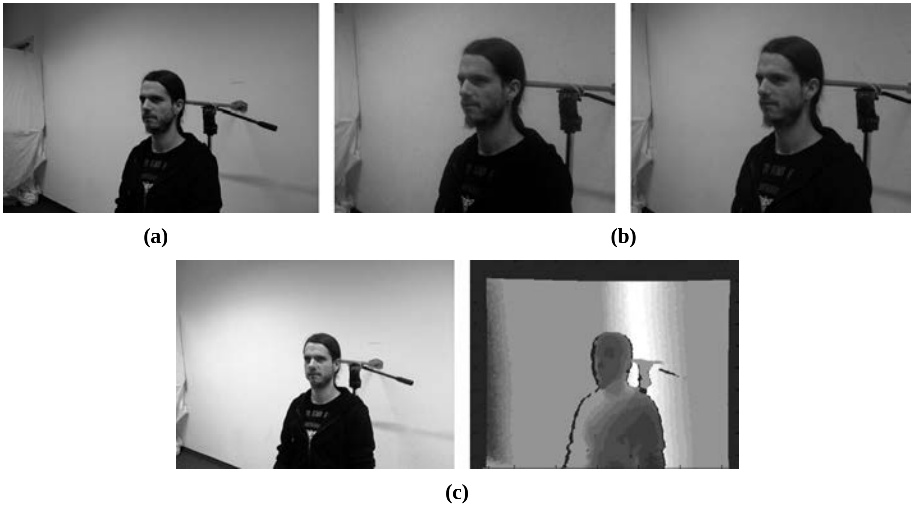

Figure 4.
2013 Hagenberg stereo vision pan shot face database with (a) high quality, colored 2D images, (b) colored 2D mobile device stereo camera image pairs and (c) OpenKinect colored 2D and range images

Table I.
Look directions and facial expressions for each pan shot

Pan shot 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	Lightly bottom right	Normal
13	Slightly bottom right	Smiling
14	Slightly bottom right	Eyes closed
15	Slightly bottom right	Mouth slightly opened
16	Slightly left right	Normal
17	Slightly left right	Smiling
18	Slightly left right	Eyes closed
19	Slightly left right	Mouth slightly opened

Plate 2.
Preprocessed images of eight different pan shots (nr 0-7), featuring different look directions and facial expressions

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:

- 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.
- 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's reliability.
- 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.

For research purposes, access to the 2013 Hagenberg stereo vision pan shot face database is available on request via e-mail, and we will be make it fully public in the near future.

5. Test setup and results

We use the high quality, colored, 2D face images from the Hagenberg stereo vision pan shot face database as input images for our tests. 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 1,000 px × 1,333 px size, with large parts of the left and right side of the image being pruned. This is done in order to:

- reduce the calculation power needed for processing the images; and
- 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 (Plate 3). As deciding if a face should still be counted as detected correctly is:

- hard to be done automated; and
- 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 II.

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.

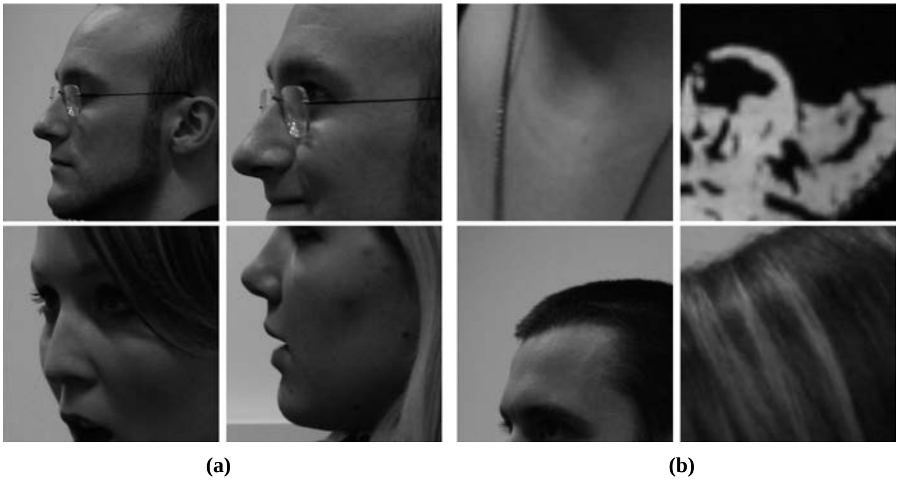

Plate 3.
Deciding on (a) still correctly detected faces and (b) already incorrectly detected faces

Table II.
Face detection results

Perspective	Nr correct	Nr incorrect	Nr not detected	Correct 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

This finding is consistent with more intensive tests of the algorithm (Douxchamps and Campbell, 2008; Santana *et al.*, 2008), 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:

- correctly detected faces only, to evaluate the face recognition performance only; and
- 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 (Findling and Mayrhofer, 2012), we used Eigenfaces for recognition (Turk and Pentland, 1991), with which we obtained an unsatisfying overall person recognition rate of 55.8 percent when applied to a preliminary face database of 38 people. Therefore, we now evaluate more promising approaches of differently configured SVM and feed forward neural networks (FFNN) as face classifiers. Before performing face recognition, we resize found faces to 128 px × 128 px 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:

- the number of hidden layer neurons for using FFNN classifiers; and
- configuring the SVM parameters corresponding to the used kernel, as suggested by Hsu *et al.* (2003).

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 percent of the data from the corresponding perspective, so that the other 40 percent 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 SVM

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.

Training and test procedure for neural networks

For each angle, subject and classifier, ten FFNN are trained with the correspondent part of the train set. 30 percent of the total set of the corresponding perspective is used for training the network, 12 percent for cross validation to stop the training if improvements become too small, and 18 percent to evaluate the ten 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 III shows the configuration of the best performing classifiers and their corresponding parameter configuration. These classifiers were evaluated once each with:

- using correctly detected faces only for training and test; and
- using incorrectly detected faces as well as correctly detected faces for training and test (Figure 5).

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 currently used face detection

Table III.
Classifier parametrization

Nr	Classifier	Kernel	Neurons	Cost	Gamma	Degree	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

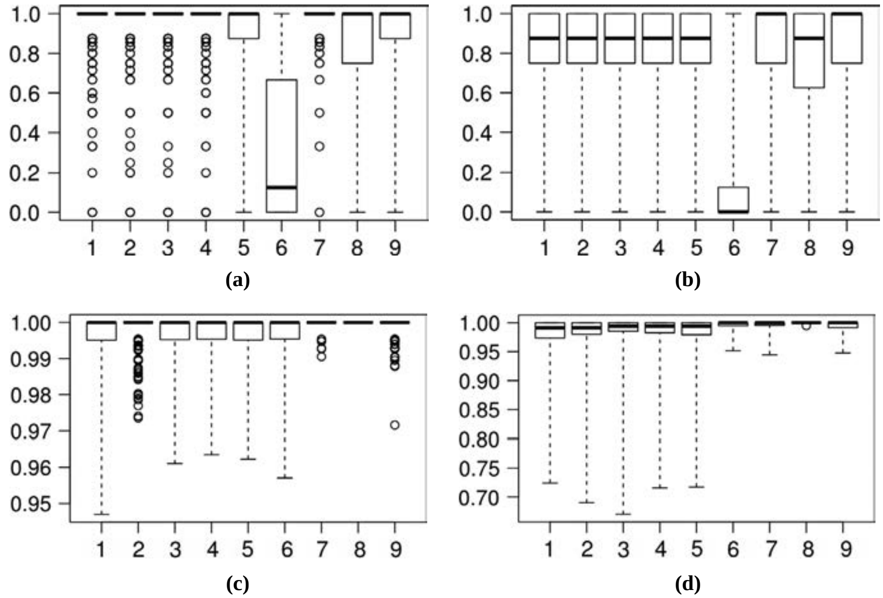

Figure 5.
Face recognition results: true positives for using (a) correctly detected faces only, (b) incorrectly detected faces as well, and true negatives for using (c) correctly detected faces only, (d) incorrectly detected faces as well

mechanism will not be sufficient. Hence, more robust and reliable approaches to finding faces in images will be included in the focus of our future research.

We achieved a true positive/negative face recognition rate of 0.9781/0.9998 when using correctly detected faces only, and of 0.8622/0.9957 when using correctly and 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 (Figure 6). 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 SVM and neural networks, to be sufficient for our current 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.

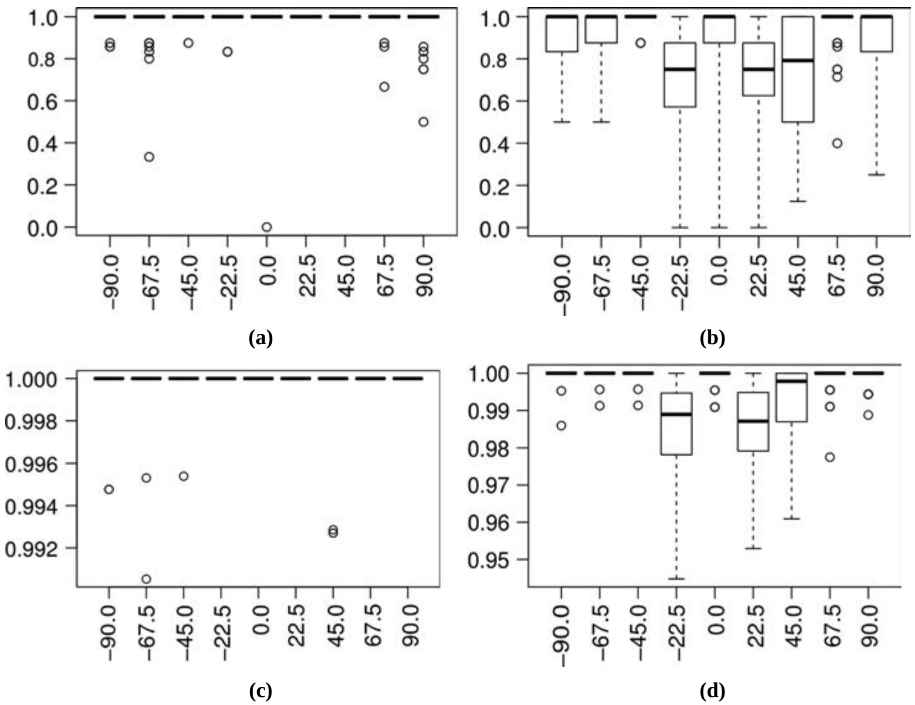

Figure 6.
Linear kernel SVM face recognition results for different perspectives: true positives for using (a) correctly detected faces only, (b) incorrectly detected faces as well, and true negatives for using (c) correctly detected faces only, (d) incorrectly detected faces as well

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.

6. Conclusion

We are working on a face unlock application that aims to use most information available to a mobile phone from a pan shot around the user's head. The system is intended to be convenient to use, reliable and harder to attack than using frontal face information only. For processing face information from different points of view, we use multiple classifiers for face detection and recognition. The disadvantage is that we require more training data. However, we argue that the significant gain in security outweighs the longer training phase, because training only needs to be performed once for each user. Metadata, obtained from sensors such as a gyroscope, is used to determine which classifier to use for which information.

Currently, 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 SVM 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 percent, 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 percent) – which consequently decreases the face recognition rate as well. Therefore, future research will deal with more robust and reliable approaches to finding faces in images from different perspectives. We further intend to use more sophisticated methods for combining the classifier results of different recording perspectives. Therefore, boosting classifier results in our pan shot face unlock toolchain will be covered in future research too.

References

Abate, A.F., Nappi, M., Riccio, D. and Sabatino, G. (2007), "2D and 3D face recognition: a survey", *Pattern Recognition Letters*, Vol. 28 No. 14, pp. 1885-1906.

Abiantun, R. and Savvides, M. (2009), "Dynamic three-bin real AdaBoost using biased classifiers: an application in face detection", *IEEE 3rd International Conference on Biometrics: Theory, Applications, and Systems* (*BTAS '09*), pp. 1-6.

Anjos, A. and Marcel, S. (2011), "Counter-measures to photo attacks in face recognition: a public database and a baseline", *International Joint Conference on Biometrics* (IJCB 2011), pp. 1-7.

Aviv, A.J., Gibson, K., Mossop, E., Blaze, M. and Smith, J.M. (2010), "Smudge attacks on smartphone touch screens", *Proceedings of the 4th USENIX Conference on Offensive Technologies*, pp. 1-7.

Bao, W., Li, H., Li, N. and Jiang, W. (2009), "A liveness detection method for face recognition based on optical flow field", *International Conference on Image Analysis and Signal Processing* (*IASP 2009*), pp. 233-236.

Belhumeur, P.N., Hespanha, J.P. and Kriegman, D.J. (1997), "Eigenfaces vs Fisherfaces: recognition using class specific linear projection", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 19 No. 7, pp. 711-720.

Bowyer, K.W., Chang, K. and Flynn, P. (2006), "A survey of approaches and challenges in 3D and multi-modal 3D+2D face recognition", *Computer Vision and Image Understanding*, Vol. 101 No. 1, pp. 1-15.

Douxchamps, D. and Campbell, N. (2008), "Robust real time face tracking for the analysis of human behaviour", in Popescu-Belis, A., Renals, S. and Bourlard, H. (Eds), *Machine Learning for Multimodal Interaction*, Springer, Berlin, pp. 1-10.

Findling, R. and Mayrhofer, R. (2012), "Towards face unlock: on the difficulty of reliably detecting faces on mobile phones", in Khalil, I. (Ed.), *Proceddings MoMM 2012: 10th International Conference on Advances in Mobile Computing and Multimedia*, ACM, New York, NY, pp. 275-280.

Frischholz, R. and Werner, A. (2003), "Avoiding replay-attacks in a face recognition system using head-pose estimation", *IEEE International Workshop on Analysis and Modeling of Faces and Gestures* (*AMFG 2003*), pp. 234-235.

Hjelmås, E. and Low, B.K. (2001), "Face detection: a survey", *Computer Vision and Image Understanding*, Vol. 83 No. 3, pp. 236-274.

Hsu, C.W., Chang, C.C. and Lin, C.J. (2003), *A Practical Guide to Support Vector Classification*, Department of Computer Science and Information Engineering, National Taiwan University, Taipei.

Hsu, R.-L., Abdel-Mottaleb, M. and Jain, A.K. (2002), "Face detection in color images", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 24 No. 5, pp. 696-706.

Huang, D., Shan, C., Ardabilian, M., Wang, Y. and Chen, L. (2011), "Local binary patterns and its application to facial image analysis: a survey", *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, Vol. 41 No. 6, pp. 765-781.

Jain, A.K. and Li, S.Z. (2005), *Handbook of Face Recognition*, Springer, Secaucus, NJ.

Lienhart, R. and Maydt, J. (2002), "An extended set of haar-like features for rapid object detection", *IEEE International Conference on Image Processing 2002*, pp. 900-903.

Liu, C. (2003), "A Bayesian discriminating features method for face detection", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 25 No. 6, pp. 725-740.

Martinkauppi, B. (2002), "Face colour under varying illumination – analysis and applications", PhD thesis, University of Oulu, Oulu.

Mayrhofer, R. and Kaiser, T. (2012), "Towards usable authentication on mobile phones: an evaluation of speaker and face recognition on off-the-shelf handsets", *Proceedings of the 4th International Workshop on Security and Privacy in Spontaneous Interaction and Mobile Phone Use, Colocated with Pervasive 2012*.

Mitchell, T.M. (1997), *Machine Learning*, McGraw-Hill, New York, NY.

Pentland, A., Moghaddam, B. and Starner, T. (1994), "View-based and modular eigenspaces for face recognition", *IEEE Computer Society Conference on Computer Vision and Pattern Recognition 1994* (*CVPR '94*), pp. 84-91.

Phillips, P.J. (1998), "Support vector machines applied to face recognition", in Jordan, M.I., Kearns, M.J. and Solla, S.A. (Eds), *Neural Information Processing Systems*, pp. 803-809.

Rowley, H.A., Baluja, S. and Kanade, T. (1998), "Neural network-based face detection", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 20 No. 1, pp. 23-38.

Sahoolizadeh, H., Sarikhanimoghadam, D. and Dehghani, H. (2008), "Face detection using gabor wavelets and neural networks", *World Academy of Science, Engineering and Technology*, Vol. 21 No. 97, pp. 552-554.

Santana, M.C., Déniz-Suárez, O., Antón-Canalís, L. and Lorenzo-Navarro, J. (2008), "Face and facial feature detection evaluation – performance evaluation of public domain haar detectors for face and facial feature detection", in Ranchordas, A. and Araújo, H. (Eds), *International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications*, pp. 167-172.

Santana, M.C., Déniz-Suárez, O., Hernández-Sosa, D. and Lorenzo, J. (2011), "A comparison of face and facial feature detectors based on the Viola-Jones general object detection framework", *Machine Vision and Applications*, Vol. 22 No. 3, pp. 481-494.

Schaub, F., Deyhle, R. and Weber, M. (2012), "Password entry usability and shoulder surfing susceptibility on different smartphone platforms", *Proceedings of the 11th International Conference on Mobile and Ubiquitous Multimedia*, ACM, New York, NY, pp. 13:1-13:10.

Schneiderman, H. and Kanade, T. (2004), "Object detection using the statistics of parts", *International Journal of Computer Vision*, Vol. 56 No. 3, pp. 151-177.

Sung, K.K. and Poggio, T. (1998), "Example-based learning for view-based human face detection", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 20, pp. 39-51.

Tari, F., Ozok, A.A. and Holden, S.H. (2006), "A comparison of perceived and real shoulder-surfing risks between alphanumeric and graphical passwords", *Proceedings of the Second Symposium on Usable Privacy and Security*, ACM, New York, NY, pp. 56-66.

Tresadern, P., Cootes, T., Poh, N., Matejka, P., Hadid, A., LeÇvy, C., McCool, C. and Marcel, S. (2013), "Mobile biometrics: combined face and voice verification for a mobile platform", *IEEE Pervasive Computing*, Vol. 12 No. 1, pp. 79-87.

Tronci, R., Muntoni, D., Fadda, G., Pili, M., Sirena, N., Murgia, G., Ristori, M. and Roli, F. (2011), "Fusion of multiple clues for photo-attack detection in face recognition systems", *International Joint Conference on Biometrics*, pp. 1-6.

Turk, M. and Pentland, A. (1991), "Eigenfaces for recognition", *Cognitive Neuroscience*, Vol. 3 No. 1, pp. 71-86.

Viola, P. and Jones, M. (2001), "Rapid object detection using a boosted cascade of simple features", *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Vol. 1, pp. 511-518.

von Zezschwitz, E., Koslow, A., De Luca, A. and Hussmann, H. (2013), "Making graphic-based authentication secure against smudge attacks", *Proceedings of the 2013 International Conference on Intelligent User Interfaces*, ACM, New York, NY, pp. 277-286.

Wagner, M. and Chetty, G. (2009), "Liveness assurance in face authentication", in Li, S.Z. and Jain, A. (Eds), *Encyclopedia of Biometrics*, Springer, Berlin, pp. 908-916.

Wechsler, H. (2006), *Reliable Face Recognition Methods: System Design, Implementation and Evaluation* (*International Series on Biometrics*), Springer, Secaucus, NJ.

Wiskott, L., Fellous, J.-M., Krüger, N. and Malsburg, C.V.D. (1997), "Face recognition by elastic bunch graph matching", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 19, pp. 775-779.

Yang, M.-H., Kriegman, D. and Ahuja, N. (2002), "Detecting faces in images: a survey", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 24 No. 1, pp. 34-58.

Yuille, A.L., Hallinan, P.W. and Cohen, D.S. (1992), "Feature extraction from faces using deformable templates", *International Journal of Computer Vision*, Vol. 8 No. 2, pp. 99-111.

Zarit, B.D., Super, B.J. and Quek, F.K.H. (1999), "Comparison of five color models in skin pixel classification", *International Workshop on Recognition, Analysis, and Tracking of Faces and Gestures in Real-Time Systems*, pp. 58-63.

Zhang, X. and Gao, Y. (2009), "Face recognition across pose: a review", *Pattern Recognition*, Vol. 42 No. 11, pp. 2876-2896.

Zou, X., Kittler, J. and Messer, K. (2007), "Illumination invariant face recognition: a survey", *First IEEE International Conference on Biometrics: Theory, Applications, and Systems* (*BTAS 2007*), pp. 1-8.

Further reading

Zhao, W., Chellappa, R., Phillips, P.J. and Rosenfeld, A. (2003), "Face recognition: a literature survey", *ACM Computing Surveys*, Vol. 35 No. 4, pp. 399-458.

About the authors

Rainhard Dieter Findling received his BSc degree in mobile computing from the University of Applied Sciences Upper Austria, where he is currently finishing his MSc degree in mobile computing. He works as a Research Associate at u'smile, the Josef Ressel Centre for User-Friendly Secure Mobile Environments. His research interests include machine intelligence and security in the context of mobile environments/ubiquitous computing. Rainhard Dieter Findling is the corresponding author and can be contacted at: rainhard.findling@fh-hagenberg.at

Rene Mayrhofer currently holds a Full Professorship for Mobile Computing at Upper Austria University of Applied Sciences, Campus Hagenberg, Austria. Previously, he held a Guest Professorship for Mobile Computing at University of Vienna, Austria, during which he received his venia docendi for Applied Computer Science. His research interests include computer security, ubiquitous computing, and machine learning, which he brings together in his research on intuitive and unobtrusive techniques for securing spontaneous interaction. He received his Dipl.-Ing. (MSc) and Dr techn. (PhD) degrees from Johannes Kepler University Linz, Austria, and subsequently held a Marie Curie Fellowship at Lancaster University, UK.