Verifying Liveness by Multiple Experts in Face Biometrics

K. Kollreider, H. Fronthaler and J. Bigun
Halmstad University, SE-30118, Sweden
{klaus.kollreider,hartwig.fronthaler,josef.bigun}@hh.se

Abstract

Resisting spoofing attempts via photographs and video playbacks is a vital issue for the success of face biometrics. Yet, the "liveness" topic has only been partially studied in the past. In this paper we are suggesting a holistic liveness detection paradigm that collaborates with standard techniques in 2D face biometrics. The experiments show that many attacks are avertible via a combination of anti-spoofing measures. We have investigated the topic using real-time techniques and applied them to real-life spoofing scenarios in an indoor, yet uncontrolled environment.

1. Introduction

Biometrics is becoming recognized all over the world, not the least through the US-VISIT traveler program and the introduction of the biometrics-enabled passports. A critical task is to guarantee the "liveness" or liveliness of what is verified by the biometric authentication system. Especially in preparation for biometrics-enabled, unattended gates and services this is a relevant issue. In this study we focus on and review anti-spoofing measures for 2D face biometrics and, to be practical we did not presume that specialized hardware (e.g. thermal camera) will be available. A face recognition system may easily be spoofed by a photograph downloaded from somebody else's homepage. The problem becomes more intricate assuming that an attacker utilizes a video, which is, given today's digital cameras and mobile devices, a realistic scenario. Obviously, (near)-infrared sensing and 3D face recognition, or multi-modal biometrics are advantageous here, too. They are however more expensive to deploy or may require substantial user collaboration to work in practice. We consider them as complementary measures to our study.

The first column of table 1 contains thinkable attacks in the category of face biometrics. The first attack implies a registered client's *face photograph* presented to the biometric system. We will refer to the second attack as *photographic mask*, since it implies that one has acquired a high resolution photograph of another person's face, cut out the eyes and the mouth, and looks through it like in a mask. The third attack happens by means of a client's *face video*, e.g. presented via a mobile/portable player to the system. Possible countermeasures are listed in the second column, where the numbers *I,II* and *III* refer to measures exploiting *eye-blinking*, *mouth movements* and *3D properties* of a real face (head), respectively, without specifying a certain technique. The third column contains the mode of the liveness detection system that is required at the least, i.e. whether it is *non-interactive* or *interactive*. Note, that we do not know the kind of attack in advance, i.e. our liveness detection system needs to assume the worst (face video) at any time, for which table 1 suggests interactive mode. Presumably, we cannot rely on eye-blinking or utterances, even on command, assuming an attacker that uses a photographic mask (therefore these measures are written in parenthesis). Are they sufficient against spoofing in biometrics [9, 3]? We have studied this issue in our experiments, which we will report further below.

Attack	Counterm.	Mode
Photograph	I,II,III	non-interactive
Ph. wrapped over face	I,II,III	-”-
+ eyes/mouth cut out	III,(I,II)	-”-
Video (mobile player)	I,II	interactive

Table 1. Attacks and Countermeasures exploiting I: Eye-Blinking, II: Mouth movements, and III: 3D properties

The novelties this study brings along are summarized below:

- We devise how to enable standard 2D face biometrics to avert known and potential spoofing attempts
- We show the real-time effectiveness of our schemes in practical scenarios

1.1. Related Work

There have been several suggestions to avert spoofing attacks via photographs, beginning with [2]. By tracking facial landmarks, the eyes and the mouth, they constructed a relative depth map via structure-from-motion. Although

978-1-4244-2340-8/08/$25.00 ©2008 IEEE

they suggested that a presented photograph could produce a constant depth map whereas a live face yields varying depths, they did not present further experimental results on liveness detection. The study of [8] suggested to analyze the Fourier spectrum of an image (in a sequence) and claimed that the spectrum of a presented photograph has lower energy in high frequencies, and that its temporal variance is lower compared to the live case. Thereby, they relied on small photographs (low resolution) and deformations caused by expression changes. Some approaches involved the speech modality ("talking face") [1, 4], because the fused audio-visual features are easier to classify into live/non-live. Within this paradigm, that targets on attacks both via photographs and videos, [3] is of interest on its own because the latter showed the possibility of recognizing utterances (digits 0-9) from lip motion only. These multimodal biometrics, although highly useful for both verification and liveness detection purposes, rely on very controlled conditions and are complementary to our study. Recently [9] proposed eye-blinking as anti-spoofing measure against presented photographs. They employed statistical models for its detection. We used the same video database for error quantification in section 3.1.

Within the suggested categorization in table 1, [2] investigated countermeasures of type III (exploiting 3D properties), whereas the talking face studies [1, 4] dealt with extended type II measures (exploiting mouth movements). The approach of [8] is assignable to all three types of countermeasures but it is unable to spatially localize the origin of temporal variations. Clearly, [9] described a countermeasure of type I (exploiting eye-blinking). We note that to the best of our knowledge, the possibility of a photographic mask attack as explained further above is a novel attacking concept not discussed elsewhere.

2. Verifying Liveness

For face biometrics, looking at table 1, we need to evaluate the 3D properties of the face in question and, check for at least either eye-blinking or mouth movements. We note that the second check could be easily adapted for both non-interactive and interactive mode. However, the security level and the interface scenarios change substantially. Biometric authentication/identification that does not take more than 3 seconds is commonly acceptable. We shall utilize this delay for liveness detection, meaning that we will track the person and employ a series of anti-spoofing measures while recognition is in progress. The ingredients in our liveness detection scheme are face (part) detection and motion estimation/tracking. In particular, motion estimation is an essential component of liveness detection since it assists in many countermeasures, for example, of type II and III as shown in [3] and [7], respectively. As a novelty we are additionally suggesting it for eye-blinking detection here, and exploit it also for face tracking.

A high-level description of the proposed algorithm for liveness detection is listed below:

1. Look for faces.
2. If a face is detected, start a timer to define the period for collecting evidence.
3. Collect evidence for the liveness of each of the faces.
4. After a period expires, verify the liveness of the face.

For step 3 we propose the following scheme:

a) Observe the 3D properties.

b) Observe eye-blinking or mouth movements (non-interactive mode)
OR Ask and check for responses at random:
Eye-blinking or utterances (interactive mode).

We note that in the non-interactive mode, it may be easier (more likely) to observe eye-blinking than mouth movements.

2.1. Implementation

We assume that a static camera continuously delivers frames (at least 15 fps) to the biometric authentication system. Besides the current frame, the previous frame is constantly available for analysis, e.g. by using a circular buffer.

For **step 1**, face detection, we suggest to employ the method introduced in [6]. It is an image-based technique for object detection that utilizes quantized angle features ("quangles"). Major advantages of that method are i) fast training, ii) no required preprocessing except for gradient estimation, and iii) the gradient being readily available in an image-scale pyramid for further tasks, e.g. for differential motion estimation. The outcome, whichever scale-invariant face detection employed, commonly are the estimated center coordinates and side lengths for every face contained in the considered image. Also, whichever face recognition technique a biometric system uses, it will first employ a quick face detection technique, i.e. liveness detection should not be treated as a black box but it should collaborate with face recognition in the targeted framework.

For tracking (**step 2**), we resort to a simple algorithm that exploits motion to refine its effective neighborhood. That is because we need the motion estimates even for liveness measurements, and by doing so we would obtain a computationally efficient liveness aware biometric recognition paradigm. When a face is detected in the current frame f_t, we would like to know its most likely counterpart in frame f_{t-1}, where t denotes an increasing frame index. Alternatively, we may ask where a detected face in the current frame is most likely to be expected in the next frame. Below, we use the current frame as reference and we estimate the motion w.r.t. the past frame. Let $\mathbf{x} = (c^x, c^y, s)^T$ denote the state of a face, with center coordinates c^x, c^y and scale s. The scale can, for example, be expressed by the fraction of the face width using the original image width as reference. We address vector components by a superscript that should not be confused with a running frame index (=subscript). Suppose that we have a detected face $\mathbf{x}_t$ at time t and we have $\boldsymbol{\delta}_t = (\delta_t^x, \delta_t^y)^T$, representing the motion vector between the center of two faces detected in the frames f_{t-1} and f_t, at our disposal. Then we can define a conditional probability density for the random vector as in equation 1a,

$$p\left(\mathbf{x}_{t-1}|\mathbf{x}_t\right) = n \cdot \exp\left(-\frac{1}{2} \cdot \mathbf{x}^T \mathbf{A} \mathbf{x}\right), \tag{1a}$$

with

$$\mathbf{x} = \mathbf{x}_t - \left(\mathbf{x}_{t-1} + (\boldsymbol{\delta}_t, 0)^T\right), \tag{1b}$$

$$\mathbf{A} = \frac{1}{\sigma^2} \cdot \begin{pmatrix} a^x & 0 & 0 \\ 0 & a^y & 0 \\ 0 & 0 & a^s \end{pmatrix}. \tag{1c}$$

Here $n = {\frac{(2\pi\sigma^2)^3}{a^x a^y a^s}}^{-1/2}$ is a normalization constant making p a probability density function. In this equation we have utilized a 3D-Gaussian with isotropic variance σ^2 to model the stochastic dynamics of $\mathbf{x}_{t-1}$. The diagonal elements of $\mathbf{A}$ translate to our a priori belief in the estimates as delivered by the face detection (directional variances). As such they are resolution independent, whereas σ^2 is resolution dependent. The term in parenthesis in equation 1b is computed assuming that $\boldsymbol{\delta}_t$ is delivered in the unit of pixel/frame. Also, the model assumes that the scale does not change abruptly between consecutive frames. We have access to the previous frame f_{t-1} in a buffer as well as the set $\mathbf{S}_{t-1} = \{\mathbf{x}_{t-1}\}$ of detected faces within it. Additionally, we will make use of a list $\mathbf{T}$ that will store the current scale, an initial time stamp, and coordinates for each tracked face[1].

After having obtained $\mathbf{S}_t$, the set of detected faces in the current frame, we employ the following tracking algorithm, where we use the $\bar{\cdot}$ notation to address a specific element that maximizes the probability mentioned.

- For each member of $\mathbf{S}_t$ do:
  - Estimate $\boldsymbol{\delta}_t$.
  - Calculate $p_{\max} = \max p\left(\mathbf{x}_{t-1}|\mathbf{x}_t\right)$ over all $\mathbf{x}_{t-1} \in \mathbf{S}_{t-1}$ via equation 1a and note the argument as $\bar{\mathbf{x}}_{t-1}$
  - If $p_{\max} > \tau$
    - If $\bar{\mathbf{x}}_{t-1}$ is not a new face associate, update $\bar{\mathbf{x}}_{t-1}$ in $\mathbf{T}$ to $\mathbf{x}_t$,
    - Else, initialize a new tracked face with $\mathbf{x}_t$ and append it to $\mathbf{T}$.
  - Else If $[p_{\max}(\bar{\mathbf{x}}) = \max_{\mathbf{x}\in\mathbf{T}} p\left(\mathbf{x}|\mathbf{x}_t\right)] > \tau$
    - Update $\mathbf{T}$ so that $\bar{\mathbf{x}} \leftarrow \mathbf{x}_t$.

In the algorithm above, a detected face cannot become a *tracked face* (=element of $\mathbf{T}$) unless detected and connected to other faces in at least two frames. This is useful to eliminate false detections. The last else-if statement has the purpose of resuming a tracking in case that the face has not been detected for a few frames, e.g. while the person shortly looked away. Furthermore, threshold τ is chosen relative to n, e.g. 15%.

We did not comment on the estimation of $\boldsymbol{\delta}_t$ also known as the motion vector field, or the optical flow. Computation of these vectors is also a prerequisite for step 3 that will be detailed further below. We suggest to employ the OFL method introduced in [7, 4] to approximate the optical flow, although we used a 2-frame variant in our experiments. Advantages of the OFL method include that it i) is efficient, and ii) it is sufficiently exact for the task. It is, like most OF estimation methods, a differential approach relying on smooth derivatives in t-dimension. We note that processing speed is really important here because we need to deal with live camera footage, as opposed to a video for which this smoothness is assured (e.g. by off-line processing). Likewise one should also reuse eligible signals from the face detection step, when useful. As a bonus of using the quangle-based face detector, we have the estimated gradients readily available at several scales of the current and previous image. This means that we can reuse them in the optical flow estimation algorithm. However, this strategy is possible to transplant also to other motion estimation methods that use the differential approach and allow scaled differentials, e.g. Horn-Schuncks [5].

Furthermore, we confine the areas for motion estimation to face sites. This suffices since we need to i) extract a robust motion vector at every face center for our tracking algorithm, and ii) process local motion for our anti-spoofing measures. Assume that a face with center (c^x, c^y) and width w, height h in the current frame f_t is detected. We denote the corresponding upper-left and lower-right vertices as $\mathbf{a}$ and $\mathbf{b}$, respectively. The motion vector field $\boldsymbol{\delta}_t$ is then estimated between f_{t-1} and f_t within the rectangular area spanned by $\mathbf{a}$ and $\mathbf{b}$. We normalize the face dimensions before estimating $\boldsymbol{\delta}_t$ to ensure a constant processing time. This means that we need to resize the image patch confined by ($\mathbf{a}$,$\mathbf{b}$) in both frames to the target size $n \cdot (w, h)$. Because we are using the quangle-based face detector, we have both frames' gradients at several resolutions readily available, thus we can pick the most appropriate level (at which $n \cdot (w, h)$ is nearest to a level-specific face size). This also enables faster multi-scale computation of the optical flow, contributing to its robustness against high velocities. Employing another object detection method, e.g. Viola-Jones' [10], would imply additional computations for gradient estimation and image transformations. For convenience, we will describe further processing within a normalized face

[1] The list $\mathbf{S}_t$ contains coordinates of faces but they are not put into correspondence with those found in $\mathbf{S}_{t-1}$.

patch with vertices $\mathbf{a} = (0, 0)$ and $\mathbf{b} = (nw, nh)$, without loss of generality. The motion vector needed for tracking could be derived from $\boldsymbol{\delta}_t(\mathbf{b}/2)$. We used a weighted average of motion vectors around the face center. Note that the motion vectors or $\boldsymbol{\delta}_t$ need to be re-scaled, e.g. $\boldsymbol{\delta}_t \leftarrow \boldsymbol{\delta}_t/n$.

For collecting evidence for liveness (**step 3**) we suggested to utilize 3D-properties, and either eye-blinking or mouth movements, hence we need to detect them. Motion estimation can assist here because 3D-properties are recoverable by so-called structure-from-motion approaches, and eye-blinking/mouth movements are obviously connected to motion. We suggest to use the following measurements that are taken within the normalized face patch at every frame t for a *tracked face*.

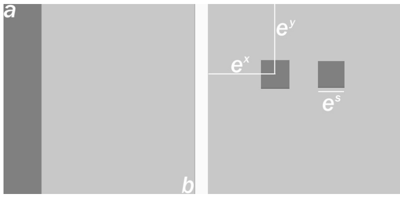

Figure 1. Masks used to extract measurements from the motion field within the normalized face patch.

a) Observing 3D-Properties:
For our first anti-spoofing measure, observing 3D-properties, we will simply extract the means and standard deviations of $|\boldsymbol{\delta}_t|$ within five equi-dimensioned vertical stripes. On the left hand side of figure 1, the dark-gray region indicates the considered area for the leftmost stripe. These five measurement pairs at the frame t are termed *rasterflow*$_t$, representing a five-tuple (2x5 vector). They encode the spatiality and are expected to converge to a wedge-pattern, abbreviated by "$\bigwedge$", when accumulated over time, in case of a live face due to its peakedness at the center compared to a photograph.

b) Observing Eye-Blinking:
Our second anti-spoofing measure is focusing on eye-blinking. The eye regions can be sufficiently well located within the face patch because they are likely to be at constant positions/scale (e^x, e^y, e^s) known from training of the object detector. This is indicated on the right hand side of figure 1, where we display the corresponding dark-gray areas that cover the two eyes. By taking the mean of $|\boldsymbol{\delta}_t|$ within these areas, and dividing it by the mean of $|\boldsymbol{\delta}_t|$ in the remaining light-gray area, we define our *eyeflow*$_t$ measurement. The latter is convenient due to the extremely high velocities at the eye sites caused by eye-blinking. Note, that when using the OFL one should only consider pixels where such velocity is non-zero for the mean calculation. This is true for the calculation of *rasterflow*$_t$ as well.

Verifying the liveness of a face after T frames, or equivalently when the timer for a tracked face expires (**step 4**) is achieved as follows: For each of the accumulated *rasterflow*$_t$ and *eyeflow*$_t$ measurements, we evaluate a liveness score, as given in equations 2a and 2b, respectively.

$$L_{3D} = \prod_{k=1}^{2} \frac{\left\langle \sum_t^T \textit{rasterflow}_t(k), \textit{wedge}(k) \right\rangle}{|| \sum_t^T \textit{rasterflow}_t(k)|| \cdot ||\textit{wedge}(k)||}, \tag{2a}$$

$$L_{Eye} = \sum_t^T I(\textit{eyeflow}_t > \tau). \tag{2b}$$

In equation 2a, *wedge* denotes a 2x5 vector containing a typical live observation and the matrices' first dimension is accessed by $(.)$. We determine its values in the experiments below. The brackets symbol $\langle, \rangle$ denotes the scalar product. In equation 2b, $I()$ denotes the indicator function that yields 1 if the argument is positive, 0 otherwise. We could observe heuristically that a reasonable value for the eyes/surroundings-flow ratio threshold τ is 3. Both scores L_{3D} and L_{Eye} are expected to be greater than zero in case of a live face, and less than or equal zero otherwise.

3. Experimental Results

In figure 2, we are displaying two typical (nonconsecutive) frames from the laboratory camera to explain our experimental setup. The functioning of the presented tracking algorithm is visualized for two tracked faces. We employed a low cost web-cam that delivered 320x240 pixel frames at 25 fps. The computing unit was a standard laptop, and C/C++ implementations for all methods were used, to provide support for feasibility. The face that is detected in the respective frame is indicated by the gray rectangle, the face that was detected in the previous frame is given by the white rectangle. Also, the motion vector $\boldsymbol{\delta}_t$ is displayed by the blue line (pointing towards the current face center = red grid's midpoint). Samples of the cond. probability density for the ideal center (2D only) of the preceding face are overlayed as a point grid, where the brightness of the points encodes the probability. The blue point is indicating the actual center position of the previous face. Points colored in

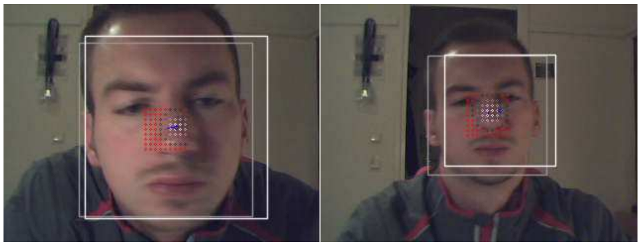

Figure 2. Two non-consecutive frames from the laboratory camera with tracking parameters overlayed (see text).

red have a probability below the threshold. The density is isotropic with a variance σ^2 of 100, i.e. $a^i = a^j = a^s = 1$. We can observe in figure 2 that the estimation of the previous face center can be perfectly achieved (left hand side), or just sufficiently (right hand side) due to abrupt movements (exaggerated). Additionally, the resilience to variation in scale and contrast changes is shown by the adverse light conditions. Both faces were successfully brought into correspondence and tracked.

We had realistic material for spoofing at our disposal, e.g. 1:1 scale printouts of portraits, unaltered and with the eyes and mouth cut out, etc., and 13 persons for representing the "live" users. The persons could move freely in front of the camera, although they were advised to look into the camera all the time. The distance to the camera in this evaluation varied between 1 and 3 meters, and the authentication delay T was assumed to be 2 seconds. Several persons (faces) could be present at a time, and multiple tracking periods were recorded for each of the users, varying also conditions. No light control was undertaken.

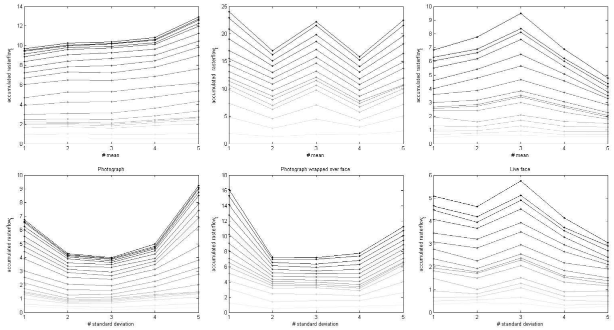

Figure 3. Accumulated *rasterflow*$_t$ means (top row) and std.-devs. (bottom) for i) photograph, ii) bent photograph, iii) live user

In figure 3, we display the evolution of $\sum_t^T$ *rasterflow*$_t$, used for score L_{3D}, split into the 5 mean values (top) and standard deviations (bottom) during the time of two seconds, i.e. for a tracked face. The darker the graph the longer the measurement has been accumulated. The leftmost column in figure 3 represents a spoofing attack by a photograph, the second column depicts an attack by a photograph wrapped over ones face (=bent photograph), whereas the third column corresponds to a "live" user. The axis labels 1..5 represent the stripes for motion extraction from left to right (compare figure 1). The mean values (top row) and corresponding standard deviations (bottom row) remain constant or are growing towards the sides because the face/photograph-background transitions contribute to motion there. In any case, the standard deviations are not growing towards the center because of the "flat" surface. The asymmetry of the graphs mirrors the real-life character of the experiments (adverse light conditions). On the other hand, we can observe two peaked graphs in the last column of figure 3, that stems from a live face. We can exploit this fact for the sought discrimination by demanding that both the mean and standard deviation graphs must jointly correlate well with a "$\bigwedge$"-template, as a consequence of equation 2a. Our experiments have shown that one can use one and the same template vector in each row of *wedge*, that is $(0, 1.5, 3.5, 1.5, 0)$ obtained by averaging all graphs originating from live faces and quantization. We want to stress that figure 3 not only represents three single examples but the general case observed in our applied experiments. Most importantly, no spoofing attempt by photographs led to $L_{3D} > 0$.

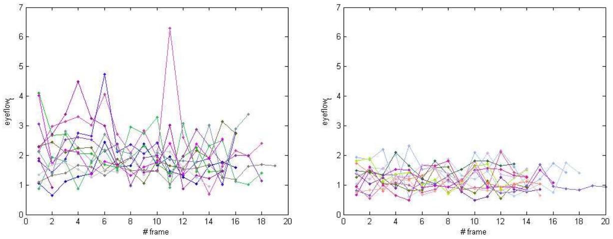

Figure 4. Measurement *eyeflow*$_t$ for tracked faces: Live cases (left) and non-live cases (right). Every color represents a different user..

In figure 4, we show the *eyeflow*$_t$ measurement for several tracked faces. Each of the color-graphs represents the ratio for a single tracking over 2 seconds, for the case of live faces (left) and spoofed faces (right), respectively. The spoofing happens as before by moving the printouts in every possible way, as well as bending them. Graphs with peak values larger than 3, representing eye-blinking, can be observed for the live faces, whereas lower fluctuation is generated for the non-live faces. Based on these observations, a liveness score $L_{Eye} = 1.4$ was calculated, using equation 2b, on the average for live faces, whereas $L_{Eye} = 0$ for non-live faces. Consequently, one can expect at least 1 eye-blinking every 2 seconds without likely rejecting actual clients. This also means that one is allowed, within an interactive system, to demand at least 1 eye-blinking at a random point without being perceived too cumbersome by the users.

When putting on the photographic mask, the situation did *not* change. This is a significant result, since one would assume that this could eliminate the eye-blinking anti-spoofing measure in whichever mode. In effect, one can only cut out small parts from the photograph in view of successful recognition. Consequently, the real eyes of the attacker can not shine through the holes due to low illumination. This is also displayed in the lower right corner of figure 5. Also, these "holes" can not generate sufficient motion because in a live face the surroundings of the eyes (or the mouth respectively), e.g. lid muscles, brows, etc. move

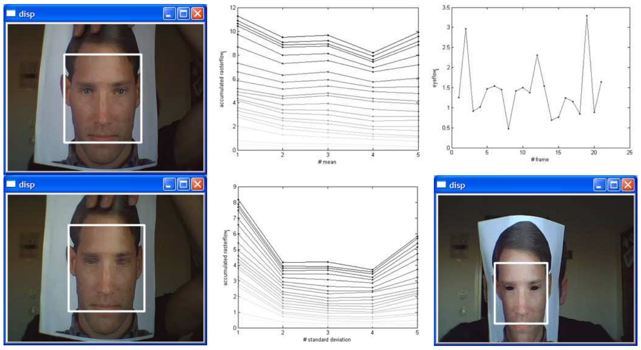

Figure 5. The lower right corner displays the photographic mask taped to ones head. All other images concern a modified attack.

as well. Furthermore, a novel experiment that is related to the photographic mask suggests the effectiveness of our 2-experts approach: Two photographs are used on top of each other, where only the foremost has cut-outs. By translating the hidden photograph one can simulate the presence of a real face plus eye-blinking (and even mouth movements to a certain extent) very well. This is displayed in the leftmost column of figure 5. We show the corresponding graphs that are used in our anti-spoofing measures, extracted during a common time frame, in the second and third column. We can observe that while the eye-blinking expert could be "successfully spoofed" to falsely accept the attack, the 3D-properties expert resists it, which ultimately allows to reject the attacker. *Accordingly, a liveness detection system relying only on eye-blinking would have been outmaneuvered by the spoofer*.

3.1. "ZJU Eyeblink" Database

For the purpose of quantifying the accuracy of the employed eye-blinking detection, we are revisiting the $\mathit{eyeflow}_t$ measurement only in this section and evaluate its accurateness on the "ZJU Eyeblink" database acquired by [9]. A modified eye-blinking expert $L'_{Eye} = \sum_t^T I_{T'}(\mathit{eyeflow}_t > \tau)$ is employed where the indicator function is taken with respect to T' frames. It means that, once $I() > 0$, we skipped counting for a number of frames that likely belong together (the same eye-blinking process). We are listing the best results in table 2. The performance is very satisfying, to the advantage of the techniques suggested here. We note that [9] employed a training approach to eye-blinking detection, with a devoted single eye localization module (which we did not), and that the data contained different camera views.

4. Conclusion

We have presented, to the best of our knowledge, the most advanced biometric spoofing attempts so far with face photographs and their combat. The suggested system is also useful against video playback attacks, since i) the attacker will likely come into the camera view field either with his real face visible or covered, and ii) eye-blinking/mouth movements on command are assumed difficult to achieve in a video. We conclude that with 2D face biometrics, *a combination of two anti-spoofing measures in an interactive scenario* is necessary for reliable liveness detection at the least. We used in-house-collected data to support our hypothesis, and one can argue about speculativeness, but there is, to the best of our knowledge, no public database for liveness quantification purposes. No data from actual crime scenes was available either. An important result is also that by collecting (accumulating) evidence during a short time we can offer a more robust liveness detection than by just analyzing, e.g. 3 frames. The performance of the employed tracking was sufficient for the task, and the suggested eye-blinking detection competed well on a public database.

	$T' \approx 16$			[9]	
Data	DR	FP(1+2)		DR	FP
		FP1	FP2		
DB1	100%	0%	0.3%	98.2%	<0.1%
DB2	97.4%	0.07%	0.1%	93.9%	<0.1%
DB3	97.6%	0%	0%	91%	<0.1%
DB4	92.7%	0.07%	0.07%	95.5%	<0.1%
Avg.	96.9%	0.04+0.12%		94.7%	<0.1%

Table 2. Detection Rates/False Positive rates on the 80 face videos from the "ZJU Eyeblink" database (20 persons ×4 sessions)

References

[1] G. Chetty and M. Wagner. Liveness Verification in Audio-Video Speaker Authentication. In *10th Australian Int. Conference on Speech Science and Technology*, p. 358–363, Sydney, Australia, December 8-10, 2004. 2

[2] T. Choudhury, B. Clarkson, T. Jebara, and A. Pentland. Multimodal Person Recognition using Unconstrained Audio and Video. In *2nd AVBPA*, Washington D.C., 22–23 March 1999. 1, 2

[3] M. I. Faraj and J. Bigun. Lip biometrics for digit recognition. In *Int. Conference on Computer Analysis of Images and Patterns*, volume 4673 of LNCS, p. 360–366, 2007. 1, 2

[4] M. I. Faraj and J. Bigun. Audio visual person authentication using lip-motion from orientation maps. *Pattern Recognition Letters*, 28(11):1368–1382, 2007. 2, 3

[5] B. Horn and B. Schunck. Determining optical flow. *Artificial Intelligence*, 17:185–203, 1980. 3

[6] K. Kollreider, H. Fronthaler, and J. Bign. Real-Time Face Detection Using Illumination Invariant Features. In *SCIA*, volume 4522 of LNCS, p. 41–50. Springer, 2007. 2

[7] K. Kollreider, H. Fronthaler, and J. Bigun. Evaluating Liveness by Face Images and the Structure Tensor. In *IEEE AutoID*, Buffalo, New York, p. 75–80, 17–18 October 2005. 2, 3

[8] J. Li, Y. Wang, T. Tan, and A. K. Jain. Live face detection based on the analysis of Fourier spectra. In *Biometric Technology for Human Identification*, p. 296–303. SPIE Volume: 5404, August 2004. 2

[9] G. Pan, L. Sun, Z. Wu, and S. Lao. Eyeblink-based Anti-Spoofing in Face Recognition from a Generic Webcamera. In *11th IEEE ICCV*, Rio de Janeiro, Brazil, October 14-20, 2007. 1, 2, 6

[10] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *IEEE CVPR*, p. 511–518, Kauai, USA, December 2001. 3