Open access • Journal Article • DOI:10.1049/IET-BMT.2012.0071

Motion-based counter-measures to photo attacks in face recognition — Source link

André Anjos, Murali Mohan Chakka, Sébastien Marcel

Institutions: Idiap Research Institute

Published on: 01 Sep 2014 - IET Biometrics (IET)

Topics: Spoofing attack

Motion-Based Counter-Measures to Photo Attacks in Face Recognition

André Anjos, Murali Mohan Chakka and Sébastien Marcel

Idiap Research Institute - Centre du Parc, rue Marconi 19, 1920 Martigny, Switzerland

{andre.anjos,murali.chakka,sebastien.marcel}@idiap.ch

Abstract

Identity spoofing is a contender for high-security face recognition applications. With the advent of social media and globalized search, our face images and videos are wide-spread on the internet and can be potentially used to attack biometric systems without previous user consent. Yet, research to counter these threats is just on its infancy - we lack public standard databases, protocols to measure spoofing vulnerability and baseline methods to detect these attacks. The contributions of this work to the area are three-fold: firstly we introduce a publicly available PHOTO-ATTACK database with associated protocols to measure the effectiveness of counter-measures. Based on the data available, we conduct a study on current state-of-the-art spoofing detection algorithms based on motion analysis, showing they fail under the light of these new dataset. By last, we propose a new technique of counter-measure solely based on foreground/background motion correlation using Optical Flow that outperforms all other algorithms achieving nearly perfect scoring with an equal-error rate of 1.52% on the available test data. The source code leading to the reported results is made available for the replicability of findings in this article.

Index Terms

Presentation Attack Detection; Counter-Measures; Face Recognition; Liveness Detection; Replay; Spoofing

I. Introduction

Identity theft is a concern that prevents the mainstream adoption of biometrics as *de facto* form of identification in high-security commercial applications [1]. Contrary to password-protected systems, our biometric information is widely available and extremely easy to sample. It suffices a small search on the internet to unveil pre-labelled samples from users at specialized websites such as Flickr or Facebook. Images can also be easily captured at distance without previous consent. Users cannot trust that these samples will not be dishonestly used to assume their identity before biometric recognition systems.

It has been suggested in the past that multimodal biometrics systems can be used to increase authentication in higher security environments [2]. However, it has been recently shown [3] that multimodal systems, if naively tunned, are intrinsically less secure than unimodal ones. This suggests each biometric modality needs to be protected by its own specialized counter-measures. In this work we are particularly concerned with direct attacks [4] to unimodal 2D (visual spectra) face-recognition systems[1]. These so-called *spoofs* [5] are direct attacks to the input sensors of the biometric system. Attackers in this case are assumed not to have access to the internals of the recognition system and manage to penetrate by only displaying biometric samples of the attacked clients to the input camera. This type of attack is therefore very easy to reproduce and has great potential to succeed [6].

Face recognition systems in particular are known to respond weakly to attacks for a long time [7], [8], [6] and are easily spoofed using one of 3 categories of counterfeits [9]: 1) a photograph; 2) a video or 3) a 3D model of the enrolled person's face. Even if attacks using videos or 3D models may seem more likely to be accepted by the recognition system, spoofs using photographs are far simpler to manufacture and should be treated with priority. It follows from reasoning that systems that are not robust to photograph attacks will also probably fail on other categories. Therefore, for the remainder of this article, we focus on spoof attempts using photographs and methods that present counter-measures to such kind of attacks based on motion artifacts.

The remaining of this text is organized as follows: Section II discusses the current state-of-the-art in anti-spoofing for 2D face recognition systems. Section III introduces the PHOTO-ATTACK database, describing its contents and usage protocols. Section IV briefly discusses three related motion-based reference systems that can be found in literature. Section V discusses our proposed Optical Flow Correlation (OFC) algorithm. Section VI reports on the experimental setup and results. Section VII analyzes results obtained on Section VI in details and suggests why OFC outperforms the three other reference systems. Finally, Section VIII concludes this article and discusses possible extensions of this work.

II. Literature Survey

Spoofing counter-measures for face recognition may be roughly classified in two categories: algorithms making use of motion or texture. Texture-based methods are very popular in face anti-spoofing possibly because of its relationship with our own discrimination capabilities on re-printed photographs. For example, in [10] the authors propose a method to detect spoofing attacks using printed photos by analyzing the

[1] We will refer to such systems simply as face recognition systems from this point onwards.

micro-textures present on the paper using a linear SVM classifier to achieve a 2.2% False-Acceptance Rate (FAR) against a 13% False-Rejection Rate (FRR) on a private database. Tan and others in [11] try to explore the Lambertian reflectance model to derive differences between the 2D images of the face presented during an attack and a real (3D) face, in real-access attempts. More recently, in [12], Määttä and others show that Local Binary Patterns (LBP) can also be used for spoofing detection achieving near perfect scoring on the PRINT-ATTACK database [13].

An advantage of texture-analysis is it relaxes requirements on anti-spoofing databases. Sets can be constructed using single shots of spoofing attacks and real-client access attempts. Motion-based spoofing detection, in turn, requires sequences of images are made available for performance evaluation. Motion analysis is therefore, less dependent on specific re-capturing artifacts explored in texture processing, being potentially able to generalize better. For example, work in [14] and [9] bring a real-time liveness detector specifically design to counter photo-spoofing using spontaneous eye-blinks (supposed to occur once every 2-4 seconds in humans). The system was evaluated on a, currently inaccessible, disjoint dataset of short video clips of eye-blinks and spoofing attempts using photographs. A later work by the same authors [15] augment the number of counter-measures deployed to include a scene context matching that helps preventing video-spoofing in stationary face-recognition systems. In [16] the authors propose a method to detect attacks produced with planar media (such as paper or screens) using motion estimation by Optical Flow (OF). Movement of planar objects is categorized as translation, rotation, normal or swing and 8 quantities that express the amount of these movements extracted from the analyzed pre-cropped face. Evaluation is carried out on a private dataset of videos to achive 90% overall classification accuracy. In [17], Kollreider and others present a technique to evaluate liveness based on a short sequence of images, also leveraging from OF analysis. The work describes a binary detector that evaluates the trajectories of select parts of the face presented to the input sensor using a simplified OF estimator followed by an heuristic classifier, with excellent results reported on a private dataset derived from the XM2VTS dataset ($\sim$1.5% equal error rate with a threshold chosen *a posteriori*).

Face anti-spoofing is not a mature field. This can be attested by the current number of publicly accessible databases. The PRINT-ATTACK database was introduced in [13]. It contains spoofing attempts with simple hard-copy print attacks recorded on video. This work also showed it was possible to use a simple motion analysis technique to obtain a good level of discrimination ($\sim$9% half-total error rate) between attacks and real-accesses. Before that, only one other dataset was publicly available: the NUAA Imposter

Database [11], which contains only single-shot photo attacks and a poorly defined protocol for training, tunning and testing of counter-measures. To explore different types of attacks without modifying other dataset parameters, we now introduce the PHOTO-ATTACK database, which enriches the PRINT-ATTACK database with mobile phone and high-resolution screen attacks.

III. The PHOTO-ATTACK Database

The PHOTO-ATTACK biometric (face) database[2] consists of short video recordings of both real-access and attack attempts to 50 different identities. To create the dataset each person recorded a number of videos at 2 different stationary conditions:

- **controlled**: In this case the background of the scene is uniform and the light of a fluorescent lamp illuminates the scene;
- **adverse**: In this case the background of the scene is non-uniform and day-light illuminates the scene.

Under these two different conditions, people were asked to sit down in front of a custom acquisition system built on an Apple 13-inch MacBook laptop and capture two video sequences with a resolution of 320 by 240 pixels (QVGA), at 25 frames-per-second and of 15 seconds each (375 frames). Videos were recorded using Apple's Quicktime format (MOV files).

The laptop is positioned on the top of a short support ($\sim$15 cm) so that faces are captured as they look up-front. The acquisition operator launches the capturing program and asks the person to look into the laptop camera as they would normally do waiting for a recognition system to do its task. The program shows a reproduction of the current image being captured and, overlaid, the output of a face-detector used to guide the person during the session. In this particular setup, faces are detected using a cascade of classifiers based on a variant of Local Binary Patterns (LBP) referred as Modified Census Transform (MCT) [18]. The face-detector helps the user self-adjusting the distance from the laptop camera and making sure that a face can be detected at most times during the acquisition[3]. After acquisition was finished, the operator would still verify the videos did not contain problems by visual inspection and proceed to acquire the next video.

[2] http://www.idiap.ch/dataset/photoattack

[3] The output of the face detector is integral part of the final dataset, so that the work described in this article can be reproduced with the same basis as used for our results.

A. Collecting samples and generating the attacks

Under the same illumination and background settings used for real-access video clips, the acquisition operator took two high-resolution pictures of each person using a 12.1 megapixel Canon PowerShot SX150 IS camera and with an iPhone 3GS (3.1 megapixel camera), that would be used as basis for the spoofing attempts. People were asked to cooperate with this part of the acquisition so as to maximize the chances of an attack to succeed. They were asked to look up-front such as in the acquisition of the real-access attempts.

To realize the photograph attacks, the operator forges an attack as described in one of the following scenarios:

- *print*: in this scenario, the operator displays hard copies of the high-resolution digital photographs that were printed on plain A4 paper using a Triumph-Adler DCC 2520 color laser printer;
- *mobile*: the operator displays photos taken with the iPhone using the iPhone screen;
- *highdef*: the operator display the high-resolution digital photos taken with the 12.1 megapixel camera using an iPad screen with resolution (1024 by 768 pixels).

The forged attacks are executed so the border of the display media is not visible when the user reproduces the spoofing attempt videos. We do this to avoid any bias on frame detection for algorithms that are developed and tested with this database. Furthermore, each video is captured for about 10 seconds in two different attack modes:

- *hand-based attacks*: in this mode, the operator holds the attack media or device using their own hands;
- *fixed-support attacks*: the operator sets the attack device on a fixed support so they don't move during the spoof attempt.

The first set of (hand-based) attacks show a *shaking* behavior that can be observed when people hold photographs of spoofed identities in front of cameras and that, sometimes, can trick eye-blinking detectors [9]. It differs from the second set that is completely static and should be easier to detect. Figure 1 shows some frames of the captured of spoofing attempts.

B. Performance Figures

A spoofing detection system is subject to two types of errors, either the real access is rejected (false rejection) or an attack is accepted (false acceptance). In order to measure the performance of a spoofing

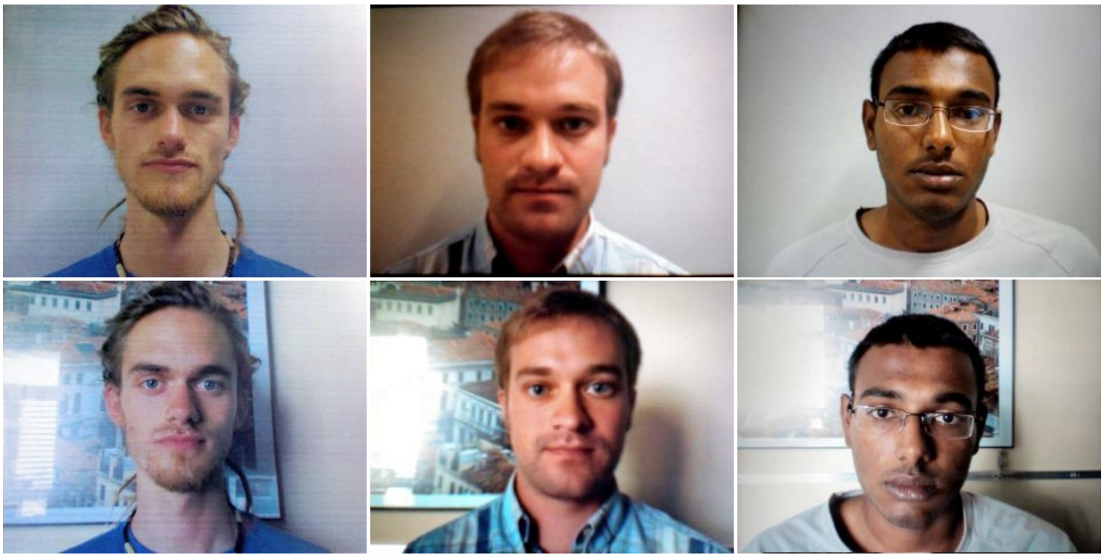


Fig. 1: Example attacks in different scenarios and with different lighting conditions. On the top row, attacks in the *controlled* scenario. At the bottom, attacks with samples from the *adverse* scenario. Columns from left to right show examples of hard-print, mobile phone and high-resolution screen attacks.

detection system, we use the Half Total Error Rate (HTER), which combines the False Rejection Rate (FRR) and the False Acceptance Rate (FAR) and is defined as:

$$HTER(\tau, \mathcal{D}) = \frac{FAR(\tau, \mathcal{D}) + FRR(\tau, \mathcal{D})}{2} \quad [\%] \tag{1}$$

where $\mathcal{D}$ denotes the used dataset. Since both the FAR and the FRR depend on the threshold τ, they are strongly related to each other: increasing the FAR will reduce the FRR and vice-versa. For this reason, results are often presented using either Receiver Operating Characteristic (ROC) or Detection-Error Tradeoff (DET) [19] curves, which graphs the FAR versus the FRR for different values of the threshold. Another widely used measure to summarise the performance of a system is the Equal Error Rate (EER), defined as the point along the ROC or DET curve where the FAR equals the FRR.

C. Protocols

The set of 800 videos (200 real-accesses and 600 attacks) is decomposed into 3 subsets allowing for training, development and testing of binary classifiers. Identities for each subset were chosen randomly but do not overlap, i.e. people that are on one of the subsets do not appear in any other set. This choice guarantees that specific behavior (such as eye-blinking patterns or head-poses) are not picked up by detectors and final systems can generalize well.

TABLE I: Number of videos in each database subset. Numbers displayed as sums indicate the amount of hand-based and fixed-support attacks available in each subset when relevant.

Type	Train	Devel.	Test	Total
Real-access	60	60	80	200
Photo-attack	90+90	90+90	120+120	300+300
Total	240	240	320	800

Moreover, each photo-attack subset can be further sub-classified into two groups that split the attacking support used during the acquisition (hand-based or fixed-support). Counter-measures developed using this database can report error figures that consider both separated and aggregated grouping, from which it is possible to understand which types of attacks are better handled by the proposed method. Table I summarizes the number of videos taken for both real-access and photo-attack attempts and how they are split in the different subsets and groups.

In the case the developed counter-measure requires training, it is recommended that training and development samples are used to train classifiers how to discriminate. One trivial example is to use the training set for training the classifier itself and the development data to estimate when to stop training. A second possibility, which may generalize less well, is to merge both training and development sets, using the merged set as training data and to formulate a stop criteria. Finally, the test set should be **solely** used to report error rates and performance curves. If a single number is desired, a threshold τ should be chosen at the development set and the Half-Total Error Rate (HTER) reported using the test set data. As means of uniformizing reports, we recommend choosing the threshold τ on the Equal Error Rate (EER) at the development set.

In the next section we investigate motion-based correlation as means to detect spoofing attacks using photographs. We start by describing three reference systems available in literature and introduce our own contribution to spoofing detection using Optical Flow Correlation (OFC).

IV. Experimental Framework

It comes rather intuitively that real access attempts should exhibit a different set of motion characteristics than photo spoofs. Real heads are 3D objects and the trajectories of each of its parts are naturally bound by the head's own shape. Photographs are, from the other perspective, static 2D objects and constrained, in this way, to other types of movement. In this section, we propose our framework for motion-based anti-

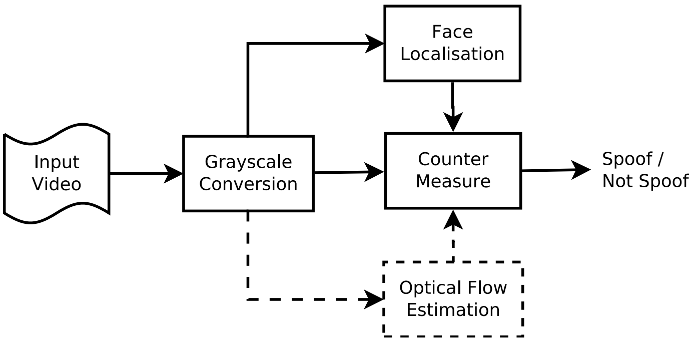


Fig. 2: Common setup for all counter-measures.

spoofing experiments used for the published results. The framework is open-source and freely distributed[4]. It allows for the reproduction of the results obtained by our newly introduced OF-based method for anti-spoofing (Section V) as well as the three reference systems selected from literature.

A. Overview

The input to motion-based algorithms for anti-spoofing is rather similar: a sequence of images as perceived by the 2D face recognition input sensor together with the output of a prefixed face detector. Figure 2 shows such a setup. The video is firstly converted to gray-scale before being fed to the face detector and, eventually, to the Optical Flow estimator unit. The gray-scaled video, the output of the face detector and the estimated flow are fed to the counter-measure that outputs scores on which it is finally evaluated.

The face detection process is not error free. In case a certain frame in the input video stream presents no detected face, we borrow the detection from any previous frame which had one. If there are no previous detections, the input is just discarded from the analysis. This procedure maximizes the number of detections available for every video sample and assures we always have valid detections when analyzing the input data.

The face detector is also subject to noise. Consecutive frames may have slightly different detected face positions. Depending on the way features are extracted from face/non-face regions, this noise may affect the counter-measure performance. To avoid this effect, comparisons are always performed by taking into consideration the face location detected on the first frame of the pair being analyzed.

Two out of the three reference systems used for comparison also use Optical Flow (OF) estimation as input for classifying real-accesses and spoofing attempts. In the past three decades, numerous attempts

[4] http://pypi.python.org/pypi/antispoofing.optflow

have been made to improve the accuracy of OF after the initial work of Horn & Schunck [20] and Lucas & Kanade [21]. As of today there exists plenty of implementations to compute the OF field for use in academic oriented research. In [22] Sun and others claim to have implemented most recent and accurate optical flow implementations in Matlab. For our experiments in this article, we have chosen to use Liu's [23] implementation for estimating dense optical flow[5], as it is developed in C++, it executes in much faster pace and could be easily ported into our unified framework. The core of the algorithm in this implementation is based on [24] and [25].

The output of the OF estimation is a velocity field ($\mathbb{O}$) that indicates both the horizontal ($\overrightarrow{U_{ij}}$) and vertical ($\overrightarrow{V_{ij}}$) components of the movement for each pixel $(x, y) = (i, j)$.

B. Reference Systems

Our open-source framework includes 3 reference systems selected from literature. They represent well prior art on motion-based anti-spoofing for face recognition. They were selected based on similarities with our own proposed algorithm input features and technique.

a) Reference System 1 (RS1) - Face center and ears: In [17], Kollreider and others describe an heuristic based on OF, which compares the movement direction between face center and the user ears to estimate whether the head is moving in a natural fashion or if the acquired imagery is derived from a user photograph. The algorithm can be summarized in these steps:

1) Find the face and ear centers on the input image;
2) Define if the head is moving more horizontally or vertically by inspecting the flow velocities at the face center;
3) Define three bounding boxes surrounding the face center and ear centers;
4) If the head is moving more horizontally, compare the thresholded average horizontal flows of the ear area to that of the face center, otherwise, use the vertical flow;
5) The sequence is considered to be a photograph if the amount of motion perceived at the ears is about the same as the motion perceived at the face center.

The comparisons are done by dividing the thresholded average flow of the face center bounding box with those of the ears to then compare its magnitude and direction as indicated on that paper. As prescribed, a sequence of images with a value of L, the *Liveness Score*, greater or equal to $\tau = 0.5$ is considered a

[5] Available from: http://people.csail.mit.edu/celiu/OpticalFlow/

live person. Note, however, that for this work we make no assumption on a threshold τ for L. Instead, we will optimize that value so as to miminize the total classification error on the development set of the PHOTO-ATTACK database when comparing this algorithm to our proposed solution.

b) Reference System 2 (RS2) - Planar object detection: Along a similar concept to the one presented in the previous section, Bao *et al.* defined at [16] a method for detecting spoofing attempts by looking for planar object movement cues in the optical flow field. Contrary to Kollreider, Bao's technique does not track specific face parts and simply divides the face area in two halves (both horizontally and vertically), trying to evaluate whether the perceived motion is derived by one of the four basic rigid object motion types: translation, in-plane rotation, panning or out-of-plane rotation (swing). An empirically derived formulation composed of averages of motion components defined L, the *Liveness Score* in this case. The higher the value of L, the more likely it is the perceived data comes from a real user trying to use the face recognition system. A low value of L indicates an attack. The original work by Bao *et al.* does not mention how the face area is obtained or how big the area around the face bounding box should be. Therefore, we consider the area of the surface to be analyzed as a hyper-parameter.

c) Reference System 3 (RS3) - Correlation with simple frame differences: In this algorithm, extracted from [13], the movement direction is ignored and one focuses on intensity only. The motion intensity in the detected face and background regions is calculated using simple gray-scaled frame-difference averages and an area-based normalization technique that removes differences in size so different face/background regions remain comparable.

To input the motion coefficients into a classifier and avoid the variability in time, we extract 5 quantities for each of the 2 regions that describe the signal pattern for windows of N images that can possibly overlap. The 5 quantities are the minimum of the signal in the window, the maximum, the average, the standard deviation and the ratio R between the spectral sum for all non-DC components and the DC component itself taking as basis the N-point Fourier transform of the signal at the window.

These quantities allow for a trained classifier to evaluate the degree of synchronized motion within the scene, during the period of time defined by N. If there is no movement (fixed support attack) or too much movement (hand-based attack), the input data is likely to come from a spoof attempt. Normal accesses will exhibit decorrelated movement between the two RoIs as normal users move independently from the background.

V. Proposed Method: Optical Flow Correlation (OFC)

Our proposed algorithm is based on similar principles as those established in RS3. It tries to detect motion correlations between the head of the user trying to authenticate and the background of the scene, which indicates the presence of a spoofing attack [13]. Instead of working with *averaged* intensities as in RS3, it uses fine-grained motion direction for deriving the correlation between these two regions. The direction of objects in the scene is estimated using Optical Flow (OF) techniques. The use of OF is expected to grant more precise estimation of motion parameters between the regions of interest in the scene, assuring that motion cues are related in direction and do not come from unrelated phenomena, as it could happen in RS3. Instead of lump-summing intensities, OFC quantizes, histograms, normalizes and directly compares motion direction vectors from the two regions of interest in order to provide a correlation score, for every analyzed frame.

We should also compare our contribution with respect to reference systems 1 and 2. The authors of RS1 and RS2 assume that a person interacting with a face recognition system will show some type of movement (either voluntarily or not) while trying to authenticate. In a scenario where no motion exists (e.g. a photograph attack with a fixed support), this assumption will fail. Yet, our newly proposed algorithm makes no such assumption: even in such a circumstance there is a strong correlation between the head and the background scene motion patterns. By doing so, it covers a wider range of possible attacks and should, therefore, show better discrimination capabilities.

OFC also introduces a new hyper-parameter that controls the amount of specific or global information that is considered while performing discrimination. As we shall explain, the number of directions Q used by the algorithm determines if the detector will observe motion patterns which may be related to specific acquisition conditions or application independent.

1) ***Feature Extraction:*** The feature extraction has 4 steps as depicted in Figure 3. The input consists of the OF horizontal and vertical velocity estimates, but also uses the face bounding boxes available in the database to separate features from the face and background regions. From those inputs, the algorithm performs the following steps:

a) We first compute the direction θ of motion for every pixel using the horizontal and vertical orientations according to a simple cartesian to polar coordinate transformation. We discard the magnitude (R) components, preserving only the movement direction θ_{ij} for every point in the original flow field as defined by $\theta_{ij} := \text{atan2}(V_{ij}, U_{ij})$ with:

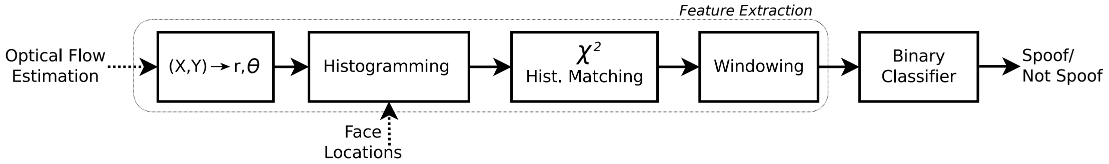


Fig. 3: Block diagram of counter-measures to spoofing attacks using correlation with optical flow.

$$\mathrm{atan2}(y,x) = \begin{cases} \tan^{-1}\left(\frac{y}{x}\right) & x > 0 \\ \tan^{-1}\left(\frac{y}{x}\right) + \pi & y \geq 0, x < 0 \\ \tan^{-1}\left(\frac{y}{x}\right) - \pi & y < 0, x < 0 \\ +\frac{\pi}{2} & y > 0, x = 0 \\ -\frac{\pi}{2} & y < 0, x = 0 \\ \text{undefined} & y = 0, x = 0 \end{cases}$$

b) The histogram computation unit calculates the normalized histograms for face and background regions solely based on the quantized angle for every flow field frame. As indicated, the face locations for every frame are supplied as input to this unit. The number of bins (Q) at the angle histogram is a hyper-parameter of this algorithm, as well as the offset used for the first bin starting angle, as depicated in Figure 4. In this way, the algorithm can be tuned to capture different kinds of movements which may be intrinsically due to attacks;

c) The next block in the feature extraction computes the χ^2 distance [26], between the angle histograms of face and background regions. If F_i is a bin value at the angle histogram of the face region, B_i is the corresponding bin value in the angle histogram of the background region and i is the bin number, then χ^2 statistic can be described mathematically as follows:

$$\chi^2(F,B) = \sum_i \frac{(F_i - B_i)^2}{F_i + B_i}. \tag{2}$$

This unit yields a single χ^2 score per computed flow field frame;

d) The windowing unit averages the χ^2 scores over a window size of N frames, with a possible specified overlap size (also in number of frames).

*2) **Classification**:* The scores computed from the windowing unit are fed to the binary classifier, which detects the spoofing attacks based on a threshold on the equal error rate (EER) tunned at the development

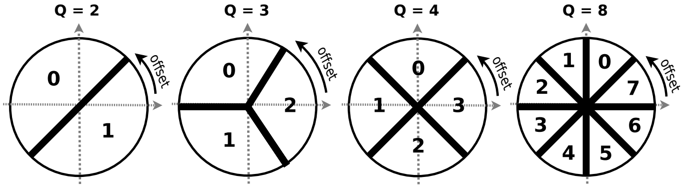


Fig. 4: Offsetting for Algorithm 4, if considering a different number of bins (Q) for the histogramming block.

set. Ideally, from Equation 2, attacks are expected to have scores close to 0 due to correlated motion between the face and the background ares. Scores of real accesses should be greater than 0 due to the fact that the face region moves independently from the background region.

VI. Experiments

This section reports on performance numbers obtained by the four algorithms described in the previous section: the three reference systems and our own contribution. The evaluation procedure is divided into two steps:

1) **Initial Tunning:** Each algorithm possesses a set of parameters that are not shared with the others. We start by optimizing each algorithm with respect to their specific parametrization using the PHOTO-ATTACK database;
2) **Common Analysis:** At a second stage, with individual parameters optimized based on the initial tunning, we vary a common set parameters that are applicable to some of the algorithms and compare, objectively, their performance.

A. Performance of Reference Systems

a) RS1 - Face center and ears: To implement this algorithm from the setup defined on Figure 2 one must derive the individual locations of the center of the face and of each ear. We have done this by hard-wiring the location of the face center (the nose tip) and the ears given the face location bounding box using average values for the human face from [27] and some empirical testing. Furthermore, the original paper by Kollreider *et al.* proposes a fixed 40×40 pixel window size for the motion estimation at the face center and a 20×40 pixel window size for the estimation of the motion around ears. In the PHOTO-ATTACK database though, faces have different sizes depending on how far (or close) they are to the input sensor. To overcome this difference we normalize the window size for the face center and ears by the size of the located face. One should also note that, while hard-wiring cannot replace precise key-point localization, the algorithm at [17] is itself not dependent on precise keypoint localization for as long as it has access to vertical edges, according to the authors. Our approximation, therefore, can be considered a reasonable assumption given the lack of software to reproduce the exact results. The final values that define the relationship between the original paper's windows and the ones used for our database are as follows:

- Face center: $(x, y) = (\text{TL}_x + 0.5W,\ \text{TL}_y + 0.395H)$;
- Face center window: instead of 40×40, $0.421W \times 0.421H$;
- Left ear center: $(x, y) = (\text{TL}_x, \text{TL}_y + 0.526H)$;
- Right ear center: $(x, y) = (\text{TR}_x, \text{TR}_y + 0.526H)$;
- Ear windows: instead of 20×40, $0.211W \times 0.421H$.

Where H and W are the height and width of the bounding box provided by the face detector we distribute with the database. TL and TR represent the top-left and top-right coordinates of the bounding box.

The only parameter that should be tunned according to [17] is the value of the threshold (henceforth denominated α). Furthermore, the authors indicate that such a value should be searched in the range $[1.0, 2.0]$, where $\alpha = 1.0$ indicates the obvious setting while values above 1.5 would further decrease the probability of false accepts. We scanned the range between 0.8 and 2.0, inclusive, with steps of 0.1 and present the results of such a counter-measure against the PHOTO-ATTACK database in Table II. Results in this table represent the development set EER and the test set HTER taking into consideration a separation threshold as indicated by the column labelled α. From $\alpha = 0.8$ until 1.5 the method works as prescribed by the reference, except for the poor performance. From $\alpha = 1.5$ onwards, we had to decrease the threshold τ to re-optimize for such aggressive values of α.

b) RS2 - Planar object detection: The only parameter that requires tunning for this method is the total surface area to be considered when deriving the liveness score L. We vary the amount of extra pixels taken from the bounding-box defined by our own face localization output, from 0 up to considering the whole available image. Table III presents these results.

Note that some of the performance figures are bracketed with parenthesis. These cases indicate that the actual scores provided by the counter-measure had to be multiplied by -1 so that results shown would be below chance. This *score inversion* is discussed in details on Section VII.

TABLE II: Optimization of the threshold α for RS1 from [17]. Values are given in % and represent the statistics over every flow field computed between any two consecutive frames for development or test sets.

α	τ **for** L	**Dev. EER**	**Test HTER**
0.8	0.38	43.79	43.37
0.9	0.38	41.44	41.15
1.0	**0.38**	**39.50**	**39.38**
1.1	0.38	40.22	40.21
1.2	0.38	42.16	42.41
1.3	0.38	44.37	44.62
1.4	0.38	46.04	46.39
1.5	0.38	47.41	47.78
1.6	0.13	44.22	44.91
1.7	0.13	45.02	45.84
1.8	0.13	45.93	46.84
1.9	0.13	46.75	47.60
2.0	0.13	47.48	48.36

TABLE III: Optimization of the total surface area when implementing the algorithm RS2, in [16]. Values are given in % and represent the statistics over every flow field computed between any two consecutive frames for development or test sets. Values inside parenthesis indicate that attacks and real-accesses had to be taken from the inverse side of the boundary L (see text).

Surface	τ **for** L	**Dev. EER**	**Test HTER**
0	**0.55137**	**(40.54)**	**(39.84)**
5	0.58562	(40.90)	(40.53)
10	0.62192	(41.56)	(41.31)
20	0.68887	(42.90)	(43.27)
50	0.81614	(47.22)	(48.61)
all image	0.87778	48.13	46.02

c) RS3 - Correlation with simple frame differences: In this algorithm, there are 2 parameters which require tunning: the window-size used for the feature extraction and the classification system. The classifier inputs 10 quantities which represent the two sets of five features from the face and background regions of the scene. The window-size parameter is an integral part of the feature extraction process of this algorithm. Therefore, to further understand the capabilities of this algorithm, we choose to jointly optimize the window size and the classification system. The overlapping O between different windows is fixed so that $O = N - 1$, where N is the window-size being probed.

For this article, we chose to use, as classifier, a multi-layer perceptron (MLPs) [28] with a feed-forward architecture containing a single hidden layer with a varying number of units. The data from the PHOTO-

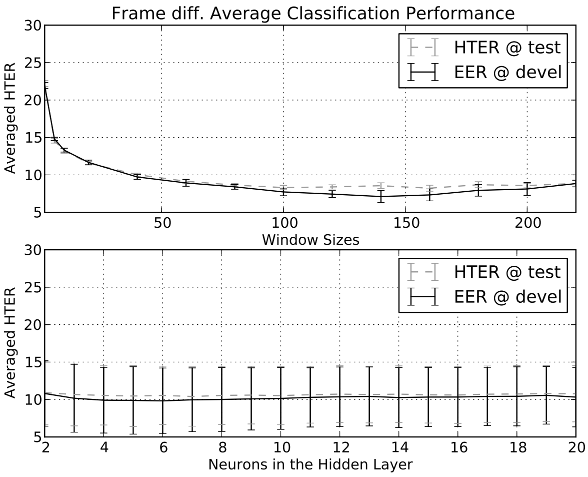


Fig. 5: Average performances for the dual optimization of RS3.

ATTACK training set is used to train the network using Resilient Backpropagation as defined in [29], with a batch size containing 500 samples picked randomly from real-accesses and attacks. Because MLP training is dependent on the initialization, we report averages and standard deviations on the EER at the development set and HTER for the test sets obtained always considering 10 training sessions with different initializations.

Figure 5 shows the average results for each of the tunned parameters. The error bars represent the standard deviation from the average HTER considering a different number of hidden neurons (top) or different window sizes (bottom). A minima is reached with about 4-6 neurons on the hidden layer for a window-size between 100 and 180 frames.

B. *Optical Flow Correlation (OFC)*

Our proposed algorithm measures motion correlation between face and background solely using the movement direction. To deploy it, one needs to tune 3 hyper-parameters: (1) the number of bins Q used for the angle histograms, (2) the amount of offset from the horizontal axis, as illustrated in Figure 4, and (3) the window-size used for the averaging processes, towards the end of the toolchain.

1) Quantization Q: Figure 6 shows the effect on the development set EER as we increase the number of bins Q used at the histogramming quantization unit from $Q = 2$ up to $Q = 8$. For these tests, we have fixed the offset used so bin 0 starts parallel to the vector $(x, y) = (1, 0)$. The windowing unit was set so averages are extracted from nearly all frames on the video sequence ($N = 220$), with an overlap of 219 ($N - 1$) frames.

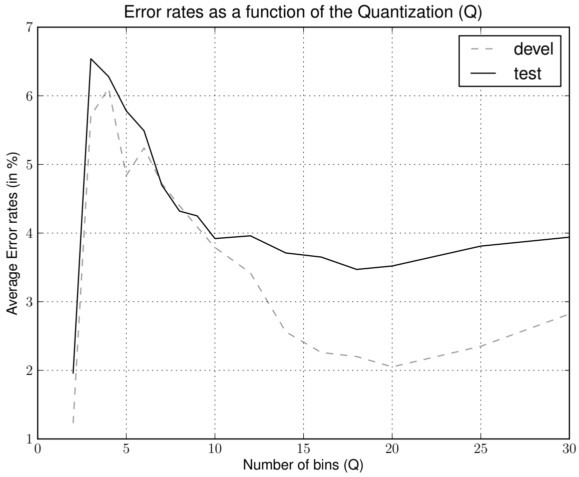


Fig. 6: Experimental results with varying number of bins Q on the PHOTO-ATTACK database for our contribution based on OFC. Fixed parameters are the Offset = 0^o, Window-size (N) = 220 frames and the Overlap = 219 frames.

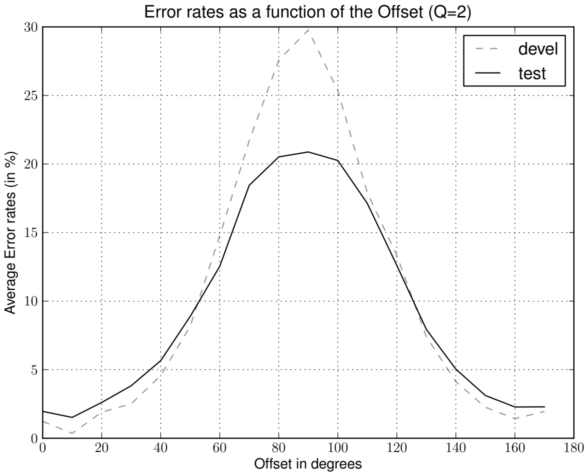


Fig. 7: Experimental results varying the offset in steps of 10^o for OFC. Fixed parameters are the number of bins in the quantization step Q = 2, Window-size (N) = 220 frames and the Overlap O = 219 frames.

Results indicate the best development EER can be achieved if one uses $Q = 2$ bins. Experiments with $Q > 30$ frames did not show any improvement in discrimination between spoofing attacks and real-access attempts.

2) Quantization Offset: We then observe how does offsetting is best tunned. Figure 7 presents the results for varying the starting offset for the histogramming procedure, in increments of 10 degrees rotating the bin 0 support vector $(x, y) = (1, 0)$ counter-clockwise, as it was described in Figure 4.

3) Window-Size: We estimate the performance impact when the ”Windowing” block (see Figure 3) of the OFC method is exposed to a window size N that varies from 2 frames to up to 220 frames. In this

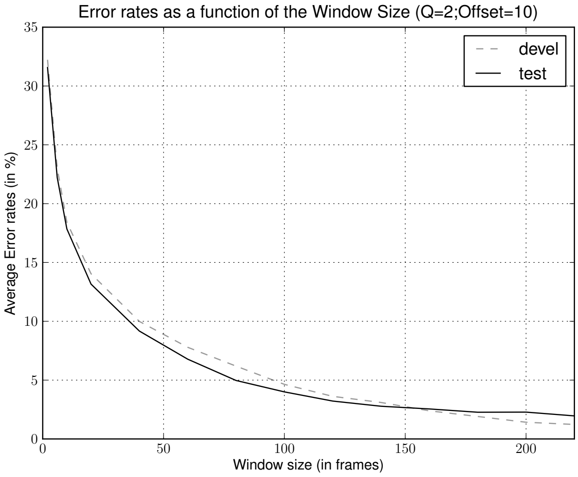


Fig. 8: Experimental results varying the window-size for OFC. Fixed parameters are the number of bins in the quantization step Q = 2, the Offset = 10^o and the Overlap O = 219 frames.

study we consider the overlap between different windows to be set to $O = N - 1$. For example, if the window size $N = 20$, the overlap is set to $O = 19$. In this way, each video with F frames generates $F - N + 1$ clusters that are considered as different observations of real-access or attacks. Results are summarized in Figure 8. It is important to note at this point that the window-size determines the amount of frames analyzed before a new score can be output by the counter-measure. It also determines the number of frames required before the first score is output by the classification scheme.

C. Common Analysis

1) Overlap: Up to this point results reported used fixed overlap sizes for windows of size N set to $O = N - 1$. In this section we analyze the effects of overlap our OFC method alongside RS3. Reducing the overlap has two consequences: firstly, it reduces the effective amount of training data available for classification tunning. Secondly, it reduces the dependencies between consecutive frame sets as the number of shared frames between such sets is reduced.

Table IV presents the results of this assessment. For this study, we assumed a window size of $N = 220$ frames for both RS3 and our new contribution based on OFC, and use the optimized parameters defined previously, only varying the amount of overlap as indicated at the first column of the table. Because of the limited number of frames in attack videos (~ 230), we avoid very low values of overlap as those will only lead to the same statistics by taking into consideration the selected window size.

TABLE IV: Average spoofing classification performance (in %) for various overlaps for a window-size of $N = 220$.

	RS3		**Proposed OFC**	
Over.	Dev. EER	Test HTER	Dev. EER	Test HTER
80	6.61 +- 1.00	7.97 +- 1.27	0.69	1.98
100	**6.53 +- 0.81**	**6.96 +- 0.88**	0.69	1.67
120	7.07 +- 0.77	8.03 +- 1.26	0.69	**1.35**
140	7.67 +- 0.74	8.04 +- 1.19	0.69	1.67
160	6.94 +- 1.00	7.40 +- 1.34	0.56	1.67
180	7.21 +- 1.01	7.55 +- 0.84	0.49	1.67
200	7.38 +- 1.00	8.07 +- 0.87	0.59	1.61
219	8.02 +- 1.59	8.27 +- 1.53	**0.36**	1.52

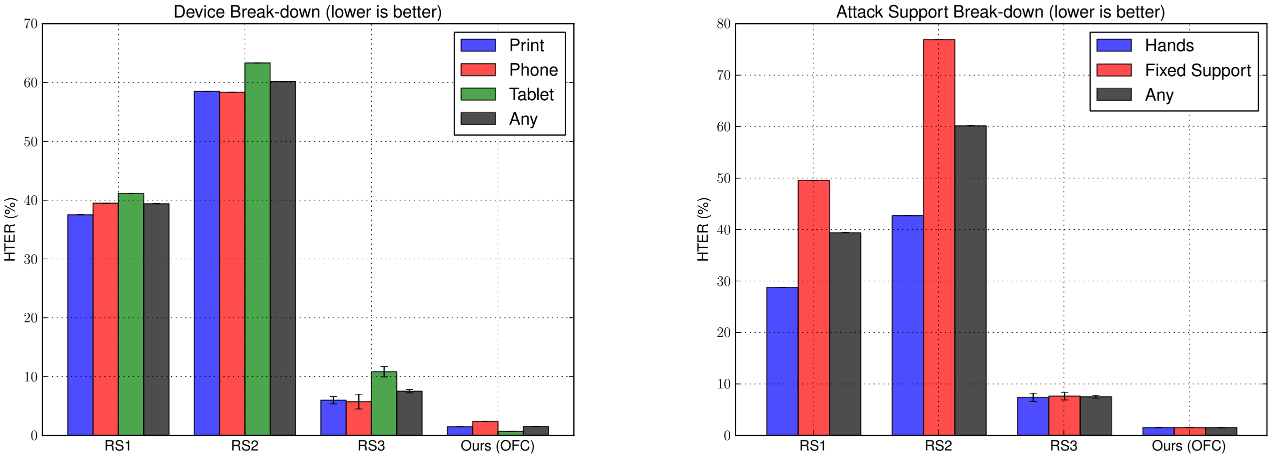
TABLE V: Average spoofing classification performance (in %) for various background-sizes (in pixels around detected face) considering optimized parametrization for every algorithm.

	RS3		**Proposed OFC**	
Bg. Size	Dev. EER	Test HTER	Dev. EER	Test HTER
5	13.07 +- 1.04	12.99 +- 1.20	19.67	16.03
10	13.35 +- 1.90	13.67 +- 1.47	9.84	9.04
20	10.24 +- 1.37	11.79 +- 1.24	4.26	2.91
30	8.15 +- 1.19	11.04 +- 0.64	2.19	2.20
40	7.46 +- 1.96	9.61 +- 0.81	0.56	2.17
50	6.75 +- 1.32	7.24 +- 0.57	**0.06**	1.76
full	**6.05 +- 0.86**	**7.52 +- 0.28**	0.36	**1.52**

2) Background Size: The background size represents the amount of background image that is utilized to extract the correlation from. Only our own OFC method and RS3 are concerned by this parametrization. To generate the values on Table V, we set parameters on the peak performance of each individual algorithm and vary the size of the background used for evaluating the correlation. Values indicated on the table represent the size of the background, in pixels, in excess of the face bounding box provided by our face detector.

D. Statistic breakdown

The attacks in the PHOTO-ATTACK database can be classified by the media used to perform the attack. Three different media types are available: hard-copy prints, mobile phones and tablets (high-definition screen). Figure 9a shows the breakdown of the best results for every algorithm taking into consideration these categories. To generate these results, we just take the values optimized for each algorithm, calculating the EER on the development set considering all possible data available. Then, we compute the HTER on



(a) Device type

(b) Attack support

Fig. 9: Breakdown by attack device and support of the HTER for the test set with threshold calculated *a priori* on the EER of the development set (in %) for various device types. Parameters for every individual algorithm are tunned to the best possible settings given results reported previously.

a subset of the test data composed only of attacks in each of the categories. Similarly, it is possible to breakdown statistics by attack support type. Two types of supports were used: attacks using the attacker's bare hands and another set using a fixed static support. Figure 9b presents the results of this second breakdown, per algorithm evaluated.

VII. Discussion

It should be noted that all the experiments described in the previous section were evaluated according to the same exact protocol and evaluation methodology. This methodology consists in optimizing any parameters from algorithms on an independent dataset referred to as development set, reporting the final performance on a separated test set. Hence, this methodology excludes algorithms are tunned favorably on data which should be exclusively used for performance evaluation.

In this section, we discuss the results presented in Section VI, along several lines: evaluation of reference systems on the PHOTO-ATTACK database or using our newly introduced OFC, the impact of the window sizes or the amount of background on newly introduced techniques, device and attack-type breakdown analysis, comparison to state-of-the-art texture-based approaches and, finally, closing remarks about algorithm complexity.

*1) **Prior State-of-the-Art:*** RS1, by Kollreider *et al.*, and RS2, by Bao *et al.*, show unexpected low discrimination capabilities when exposed to the PHOTO-ATTACK database. In particular, RS1 does not seem to tune as advertised on the original paper. One of the reasons for this degradation possibly lie in the way the face center and ears have been calculated. The second possible incompatibility can be found on the different method used for the Optical Flow estimation. These results suggest that such a technique may not generalize well to other databases or different motion estimation algorithms. The results in Table II indicate the optimal value for the threshold τ is found to be 1.0, yielding a test HTER of 39.4%. The breakdown in Figures 9a and 9b for this algorithm point out that it discriminates better hard-copy print attacks performed by hand, but fails to detect well other types of attacks. The error on the detection of fixed-support attacks is almost twice that of hand-based ones. This sum of factors indicate a possible bias on the original algorithm conception.

RS2, by Bao *et al.* seems to have its peak discrimination capabilities if applied only to the face region, as shown by the results in Table III. Error rates at this operating point are of about 40%. It is interesting to note that the threshold for L, as prescribed in the original article, does not separate real accesses and photo attacks in the same way - we had to consider the separated ranges in inverse order. The explanation for this effect may be understood by looking at the breakdown on Figure 9b. There, one can verify this algorithm only works as advertised if attacks are performed with one's own hands once more. Indeed, the working hypothesis for both RS1 [17] and RS2 [16] is that real clients trying to authenticate *display* motion differently than photo attacks. In a scenario where clients are still or have unperceivable motion and attacks are drawn from fixed stands, that assumption can no longer be true. In such a scenario, the performance of RS1 is about chance. Because RS2 makes even stronger assumptions about the motion characteristics for real-accesses, results for this system are even worse, peaking more than 75% HTER on Figure 9b.

The best results for RS3, based on simple frame differences, show that an MLP network with $\sim$5 hidden neurons achieves optimal discrimination capabilities, with $\sim$7.2% HTER on the test set, with a threshold chosen *a priori* on the development set (see Figure 5). The best performance markings are not achieved for very large window sizes, as is the case for our contribution based on OFC. The dual-optimization shown at Section VI-A indicates a minimum is reached gathering features in sets of ~ 140 frames. This apparent *saturation* of the method is possibly related to its simplistic feature extraction process, that averages motion intensity through the frames. As more frames accumulate, subtle motion differences between head and scene background are averaged out, saturating the system performance. Our proposed OFC method calculates scores on a frame-by-frame basis and uses a late score fusion to finally evaluate the scene. This technique is able to leverage better subtle motion differences through out the whole duration of the clip.

*2) **RoI-based Optical Flow Correlation**:* The newly introduced OFC algorithm, that correlates face and background motion using direction, shows the best results in this article. The system can achieve error rates of less than 2.0% if optimized adequately, while not performing badly in suboptimal operating points. It is interesting to observe, in Figure 6, that the number of bins Q used for the histogram correlation is a crucial parameter, as well as the choice of the initial histogramming offset as shown on Figure 7 (see also Figure 4). The main reason for this effect lies on the way users adequate themselves to be identified by face recognition systems: positioning their heads to fit on masks pre-disposed on the screen. In the attempt to fit, the user will move its head forward and backward, slightly. This behavior will be even more present in attacks, where the attacker will try to fit the head on the photograph to the available mask, zooming himself in and out of the detector focus. OFC can best benefit from this information only if $Q = 2$ bins are used for the histograms as it can clearly capture such a *self-adjusting effect* if the offset is ~ 0. As the offset is increased (see Figure 7), this self-adjusting discriminative information is distributed among a higher number of histogram bins and averaged out. The test HTER varies from $\sim 1.5\%$ to $\sim 20.9\%$ in the range of different thresholds which are possible. If the number of bins Q is increased to 3, the system looses a fraction of its capacity to detect the adjustment effect that is more evident in attacks and starts making use of a simpler correlations. As one increases the value of Q, the system tends to recover its original discrimination power by exploring overall scene correlation aspects. Results show a performance of 3.5% HTER (on the test set), for $Q = 20$ and the new OFC-based algorithm in that case. An increase in the number of bins Q for the histogram quantization step reduces the impact of offsetting, with a small increase in average error rates (see Figure 6).

The optimal value for Q and the offset are, therefore, application specific. If Q is small, the system better captures specific motion patterns which can be present on the current authentication setup. At those values of Q, the offset may play a role in determining the optimal direction that should be observed for maximally discriminating spoofing attacks. As Q is increased, the system moves on to capture global correlation effects and becames independent of the offset.

*3) **Window-Size Analysis**:* The analysis in Figure 8 matches intuition: as the amount of data input through the method increases, the algorithm's performance improves. RS3's performance flattens out at around 140 frames and adding more frames does not seem to improve the discrimination capacity. Interestingly, our OFC algorithm seems to be able to handle as much data as it can get with a performance that continues to improve steadily after 140 frames to reach an almost perfect scoring for the EER on the development set, below 0.4%, with a low test HTER of $\sim 1.5\%$. Overlapping (see Table IV) does not seem to change in any way these algorithms' capabilities and it is therefore advised to always choose the maximum overlap which will maximize the total amount of data which can be used for training as well as threshold setting. As a side note, experiments with windowing/overlapping for RS1 and RS2 did not show any improvement on the overall performance of these methods.

*4) **Background Wideness**:* Changing the amount of background used in the correlation procedure does affect RS3 or our OFC-based method, as shown in Table V. These are expected results since the cues for the correlation evaluation will become less and less evident as the amount of data surrounding the detected face is reduced. It is interesting to note that a reasonable performance can be achieved with about 20-30 pixels for RS3 and our OFC-based algorithm with about 10% and 2.2% test HTER respectively. This allows for the use of this technique of spoofing detection in environments which do not necessarily contain a static background.

*5) **Breakdown Analysis**:* The breakdowns (Figures 9a and 9b) also show interesting results for the RS3 and our OFC-based method. RS3 seems to work equally well for both print and photo attacks while performing poorly when the attacker uses a higher-definition screen. Our new OFC-based method has a flatter performance through the attack device types, but performs best exactly where RS3 fails. While using tablets as attack devices, the outcome is a sharper image which allows the Optical Flow estimator to perform better and our new method to better detect the correlation between the face region and the background. For the same reasons, phone attacks perform worst for it. While the decay in sharpness affects the estimation of direction and the correlation calculation, it helps the magnitude-based approach of RS3 identifying spoofing attempts. Both algorithms seem to be insensitive to the type of support used in the attack, showing evident advantage when compared to the two other reference systems.

*6) **Comparison to Texture Approaches**:* A recently organized competition on a subset of the PHOTO-ATTACK database called PRINT-ATTACK [30] showed that texture-based methods could potentially beat our proposed method under those conditions. To estimate how well such methods can work against the PHOTO-ATTACK database, we evaluated the work of Chingovska and others [31], a variant of the competition winner based on Local Binary Patterns (LBP), and to which we have access to an open-source implementation[6]. The best results that can be generated with such a package reach ∼16% HTER on the

[6] http://pypi.python.org/pypi/antispoofing.lbp

TABLE VI: Experimental error rates (in %) for OFC for various frame skip values, on the PHOTO-ATTACK database. Bins = 2. Offset = 10^o. Window-size = 220. Overlap = 219.

Skip	**Dev. EER**	**Test HTER**	$\sim$ **Required Rate (Hz)**
0	0.36	1.52	25
1	1.26	1.49	12.5
2	1.64	1.53	8.3
3	2.94	2.37	6.3
4	3.64	2.97	5
5	2.81	2.95	4.2
10	7.81	5.86	2.3
25	16.87	10.70	1
50	11.32	17.83	0.5

test set, using a combination of uniform LBP codes, principal component analysis and linear discriminant analysis.

*7) **Complexity**:* All systems developed in this article are extremely simple, mostly relying on heuristics to counter spoofing attacks. One must note, however, that RS1, RS2 and our newly introduced technique based on OFC, require the computation of the Optical Flow (OF) field, which can be considered a complex task. In our setup, for example, we could obtain a throughput of a few frames per second with the toolbox downloaded from [23]. In contrast, the frame-by-frame differences required by RS3 can be calculated using only integer operations which should, therefore, require much less computing power. The trade-off seems to affect the classification of such a method: while our method can achieve smaller error rates with a simple classification metric, RS3 must find higher-order correlations on the input feature space (through the MLP) to compensate for the simpler feature extraction.

Table VI reports the HTER for the PHOTO-ATTACK database test set with threshold chosen *a priori* on the development set when one considers the estimated flow for all frames (first row) or when we decide to skip the calculation of the OF fields for a given number of frames for a given video sequence, while keeping all other parameters fixed. For example, the second row of Table VI indicates what would happen if we skipped the calculation of the OF for every other frame in the sequence. The second row, shows what happens if we calculated the OF for only 1 frame in every 3 frames, and so on. The fourth column of such a table shows the (approximate) required OF calculation rate in Hertz. As can be seen, results are still satisfactory and better than RS3, if one computes the OF field for 1 in every 11 frames. The required processing rate in this case would be about 2.3 Hz.

One must note, nevertheless, Optical Flow estimation is still an active research field in computer vision and computing power never ceases to grow. Methods for real-time flow estimation which trade-off accuracy for execution time could be considered in place of the method used in this work.

VIII. Conclusion and Extensions

One of the easiest ways to spoof a 2-D face recognition system is by the use of photographs of attacked identities. This problem has been understood for nearly a decade now and, yet, no consensus seems to exist on techniques or best-practices to avoid this. Literature is scarce and results are difficult to reproduce. To remedy this aspect, we made public a PHOTO-ATTACK dataset and explain how to use its companion protocol. The database is sufficiently large and contains a diverse set of spoofing attacks under different conditions.

Under the light of this new dataset, we conducted experiments demonstrating previous attempts to solve the issue do not generalize well, hintting on possible reasons of why their basic assumptions fail. Yet, we also show photo attacks can be detected accurately by solely using direction-based motion correlation. With this technique we can obtain $\sim$1.5% HTER on the database test set with a threshold chosen a priori on the development set. Our newly introduced method requires the estimation of Optical Flow (OF) fields or other direction oriented features on the target images, but can dramatically improve the accuracy of spoofing detectors. This seems to generalize well for different types of attacks, using different media and support. Requirements on stationarity may be lifted by trading-off for smaller background windows that limit the area around the face to the smallest possible set for a given condition. It is also possible to trade-off processing time and accuracy by reducing the size of time-windows to which counter-measures are applied to. The software suite leading to the figures published in this article was made open-source and publicly available so our results can be easily reproduced.

The evaluated algorithms are strongly dependent on the results of a face detector framework, that prefixes our setup. The detector based on [18] detected a valid face in about 99.3% of the frames available in the PHOTO-ATTACK database. At this time, we have not evaluated the impact of undetected faces in the overall performance of the algorithm or how a change in face detection strategy (for example, using a tracker) could impact final results. This seems a natural extension to this work.

We shall continue to improve the PHOTO-ATTACK database with more challenging spoofing attempts such as those using videos of users. Additionally, more challenging conditions with poor lighting or non-stationary backgrounds should be addressed on the next iterations. On the side of detecting attacks, the fusion of both magnitude and direction cues could be studied to yield an even more robust spoofing classifier. At this point, the impact of using different Optical Flow estimation techniques could help determine our proposed algorithm's dependence on that pre-processing stage.

Finally, the scores obtained from our frame-based analysis could be merged for each video stream and the quality of counter-measures evaluated as time passes. At this stage, one may consider simple fusion schemes or more complex ones that may depend on independent quality measurements.

Acknowledgments

The authors would like to thank the Swiss Innovation Agency (CTI Project Replay) and the FP7 European TABULA RASA Project (257289) for their financial support.

References

[1] S. A. C. Schuckers, "Spoofing and anti-spoofing measures," *Information Security Technical Report*, vol. 7, pp. 56–62, 2002.

[2] T. Sun, Q. Li, and Z. Qiu, *Advances in Biometric Person Authentication*. Springer, October 2005, ch. A Secure Multimodal Biomeric Verification Scheme, pp. 233–240.

[3] R. N. Rodrigues, L. L. Ling, and V. Govindaraju, "Robustness of multimodal biometric fusion methods against spoof attacks," *Journal of Visual Language and Computing*, 2009.

[4] J. Galbally, C. McCool, J. Fierrez, S. Marcel, and J. Ortega-Garcia, "On the vulnerability of face verification systems to hill-climbing attacks," *Pattern Recognition*, vol. 43(3), pp. 1027–1038, 2010.

[5] A. K. Jain, P. Flynn, and A. A. Ross, Eds., *Handbook of Biometrics*. Springer-Verlag, 2008.

[6] N. M. Duc and B. Q. Minh, "Your face is not your password," in *Black Hat Conference*, 2009.

[7] L. Thalheim, J. Krissler, and P.-M. Ziegler, "Body check: Biometric access protection devices and their programs put to the test," *Heise Online*, 2002.

[8] B. Toth, "Biometric id card debates," *Newsletter Biometrie*, 2005.

[9] G. Pan, Z. Wu, and L. Sun, "Liveness detection for face recognition," *Recent Advances in Face Recognition*, pp. 109–124, December 2008.

[10] J. Bai, T. Ng, X. Gao, and Y. Shi, "Is physics-based liveness detection truly possible with a single image?" in *International Symposium on Circuits and Systems*. IEEE, 2010, p. 3425–3428.

[11] X. Tan, Y. Li, J. Liu, and L. Jiang, "Face liveness detection from a single image with sparse low rank bilinear discriminative model," *Computer Vision ECCV 2010*, vol. 6316, pp. 504–517, 2010.

[12] J. Määttä, A. Hadid, and M. Pietikäinen, "Face spoofing detection from single images using micro-texture analysis," in *International Joint Conference on Biometrics (IJCB)*, 2011.

[13] 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.

[14] G. Pan, L. Sun, Z. Wu, and S. Lao, "Eyeblink-based anti-spoofing in face recognition from a generic webcamera," *IEEE 11th International Conference on Computer Vision (2007)*, pp. 1–8, 2007.

[15] G. Pan, L. Sun, Z. Wu, and Y. Wang, "Monocular camera-based face liveness detection by combining eyeblink and scene context," *Journal of Telecommunication Systems*, 2009.

[16] W. Bao, H. Li, N. Li, and W. Jiang, "A liveness detection method for face recognition based on optical flow field," in *2009 International Conference on Image Analysis and Signal Processing*. IEEE, 2009, pp. 233–236.

[17] K. Kollreider, H. Fronthaler, and J. Bigun, "Non-intrusive liveness detection by face images," *Image and Vision Computing*, vol. 27, no. 3, pp. 233–244, 2009.

[18] B. Froba and A. Ernst, "Face detection with the modified census transform," in *IEEE International Conference on Automatic Face and Gesture Recognition*, 2004, pp. 91–96.

[19] A. Martin, G. Doddington, T. Kamm, M. Ordowski, and M. Przybocki, "The det curve in assessment of detection task performance," in *Fifth European Conference on Speech Communication and Technology*, 1997, pp. 1895–1898.

[20] B. K. P. Horn and B. G. Schunck, "Determining optical flow," *Artificial Intelligence*, vol. 17, pp. 185–203, 1981.

[21] B. D. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," in *Seventh International Joint Conference on Artificial Intelligence*, 1981, pp. 674–679.

[22] D. Sun, S. Roth, T. Darmstadt, and M. J. Black, "Secrets of optical flow estimation and their principles," in *IEEE Conf. on Computer Vision and Pattern Recog., CVPR,*.

[23] C. Liu, Ed., *Beyond Pixels: Exploring New Representations and Applications for Motion Analysis*. Doctoral Thesis. Massachusetts Institute of Technology, May 2009.

[24] A. Bruhn, J. Weickert, and C. Schnörr, "Lucas/kanade meets horn/schunck: Combining local and global optic flow methods," *International Journal of Computer Vision*, vol. 61, pp. 211–231, 2005.

[25] T. Brox, A. Bruhn, N. Papenberg, and J. Weickert, "High accuracy optical flow estimation based on a theory for warping," in *European Conference on Computer Vision (ECCV)*. Springer, 2004, pp. 25–36.

[26] T. Ahonen, A. Hadid, and M. Pietikäinen, "Face recognition with local binary patterns," in *European Conference on Computer Vision (ECCV)*, 2010, pp. 469–481.

[27] G. Leslie and M. Farkas, Eds., *Anthropometry of the Head and Face*. Raven Pr, 1994.

[28] C. M. Bishop, *Pattern Recognition and Machine Learning (Information Science and Statistics)*, 1st ed. Springer, October 2007.

[29] M. Riedmiller and H. Braun, "A direct adaptive method for faster backpropagation learning: the rprop algorithm," in *IEEE International Conference on Neural Networks*, vol. 1, no. 3. IEEE, 1993, pp. 586–591.

[30] M. M. Chakka, A. Anjos, S. Marcel, R. Tronci, D. Muntoni, G. Fadda, M. Pili, N. Sirena, G. Murgia, M. Ristori, F. Roli, J. Yan, D. Yi, Z. Lei, Z. Zhang, S. Z. Li, W. R. Schwartz, A. Rocha, H. Pedrini, J. Lorenzo-Navarro, M. Castrillón-Santana, J. Määttä, A. Hadid, and M. Pietikäinen, "Competition on counter measures to 2-d facial spoofing attacks," in *International Joint Conference on Biometrics (IJCB)*, 2011.

[31] I. Chingovska, A. Anjos, and S. Marcel, "On the effectiveness of local binary patterns in face anti-spoofing," in *IEEE BioSIG*, August 2012.