Face-Spoofing 2D-Detection Based on Moiré-Pattern Analysis

Diogo C. Garcia, *Member, IEEE,* and Ricardo L. de Queiroz, *Senior Member, IEEE*

***Abstract*—Biometric systems based on face recognition have been shown unreliable under the presence of face-spoofing images. Hence, automatic solutions for spoofing detection became necessary. In this paper, face-spoofing detection is proposed by searching for Moiré patterns due to the overlap of the digital grids. The conditions under which these patterns arise are first described, and their detection is proposed which is based on peak detection in the frequency domain. Experimental results for the algorithm are presented for an image database of facial shots under several conditions.**

***Index Terms*—Biometrics, face-spoofing detection, face recognition.**

I. Introduction

In recent years, researchers have devoted great attention to biometric systems and its many challenges, such as security evaluation and vulnerability assessment. In particular, face recognition systems have been widely studied, as they offer a simple and effective method for human authentication, requiring only regular cameras and specialized software [1].

Face recognition systems, however, rely on flat images in order to detect people, so that they can be easily spoofed by printed photographs and mobile displays. Several face spoofing-detection methods have been proposed [2]–[19]. Some of the approaches include image-quality analysis, motion analysis, texture analysis, or a combination of these.

Among the image-quality approaches, Li *et. al.* define a high-frequency descriptor and a lower threshold to differentiate regular and face-spoofing images [2]. Zhang *et. al.* detect printed and displayed face-spoofing images by using a support vector machine (SVM) to search for a lack of high-frequency information [3]. Tan. *et. al.* search for implausible illumination changes and for low image quality in the detected-face image [4]. Peixoto *et. al.* also search for low image quality, but also adapt input images subject to bad illumination conditions, based on contrast-limited adaptive histogram equalization [5]. Galbally *et. al.* search for low image quality using 25 general image quality features extracted from one image, such as peak-signal-to-noise ratio and the structural similarity index [6].

Anjos and Marcel search for motion inconsistencies in sequences of detected faces by comparing features in the detected-face region and the rest of the image, such as mean and standard deviation [7]. The comparison is carried using a multi-layer perceptron classifier. De Marsico *et. al.* employ a set of facial points in different frames in order to exploit geometric invariants [8]. Kollreider *et. al.* compare the motion of different parts of the detected face, which will be more diverse on a real, tridimensional face [9].

Among the texture analysis approaches, Pinto *et. al.* search in the frequency domain for noise signatures in time generated by the recaptured video to distinguish between fake and valid access [10]. Määttä *et. al.* search for texture inconsistencies in the detected-face image by applying local-binary-pattern codes on a block basis, calculating the histograms and classifying the information with a SVM [11]. Chingovska *et. al.* build upon the previous work by testing different classifiers [12]. Bharadwaj *et. al.* apply texture analysis to motion-magnified sequences [13],[14]. Kim *et. al.* analyse texture by mixing information of the power spectrum with local binary patterns [15]. Pereira *et. al.* analyse spatial and temporal textures using local binary patterns in the X-Y, X-T and Y-T dimensions [16]. Bai *et. al.* calculate the histogram of gradients of the specular component of the detected-face image in order to detect face-spoofing images [17].

Also, Schwartz *et. al.* analyse diverse spatial and temporal features, such as the histogram of oriented gradients, color frequency, gray level co-occurrence matrix, and histograms of shearlet coefficients, and integrate these features with a weighting scheme based on partial least squares [18]. Pereira *et. al.* employ two previous methods in order to detect face-spoofing images and videos [7],[12],[19].

Even though these references obtain great success in detecting face-spoofing images, they generally rely on highly empirical methods, so that replicating these results in different circumstances can be very difficult. SVMs, for instance, require balanced training databases suited for the conditions under which they are supposed to work [20]. In this paper, face-spoofing 2D detection is proposed by searching for Moiré patterns, which can be theoretically modeled as the overlap of the digital grids in the face-spoofing display and in the face-recognition camera. Unlike Li *et. al.* [2] and Zhang *et. al.* [3], who assume that face-spoofing images contain fewer high-frequency content than regular face images, we search for distinct patterns that are high-pass in nature. Like Zhang *et. al.* [3], we employ Difference-of-Gaussians (DoG) filters, but with a different objective: isolating high-frequency patterns in the frequency domain. Our approach is simpler than searching for textures in the spatial domain, as it does not require large training databases for descriptors such as local binary patterns. A theoretical analysis is first presented in order to define

D. C. Garcia is with the Department of Electronics Engineering, University of Brasilia at Gama, Brasilia 70919, Brazil (e-mail: diogogarcia@unb.br).

R. L. de Queiroz is with the Department of Computer Science, University of Brasilia, Brasilia 70919, Brazil (e-mail: queiroz@ieee.org).

the conditions under which the Moiré patterns arise, followed by the algorithm proposed for the detection of face-spoofing images. Experimental results for the algorithm are presented for a database of face images shot under several conditions.

II. Digital artifacts on face-spoofing images

Face recognition systems can be easily spoofed by images of trusted users on printed photographs or on mobile displays. However, as these spoofing techniques rely on digital media, as opposed to the analog reality of the trusted user, the digital grid of the face-recognition system camera overlaps with the grid of the digital media. In the case of printed photographs, the image grid of the camera overlaps with the printing halftoning dots, and in the case of mobile displays, it overlaps with the pixel grid. The proposed method searches for artifacts due to the overlapping of digital grids.

Figure 1 shows one of the most common artifacts of this kind: Moiré patterns [21]-[24]. Figure 1(a) shows a portion of test image *Lena*, and Fig. 1(b) is a photograph of (a), captured from a 13-inch display of a Macbook Pro using an iPhone 4 camera, without any compression artifacts. Figures 1(c) and (d) show details of Figs. 1(a) and (b), respectively, illustrating the patterns that occur after an image is recaptured from a screen. Note that these patterns are not present in the original image in Fig. 1(c).

The detection of Moiré patterns at the spatial domain can be very complex, since there is no *a priori* method to distinguish this kind of pattern from any other. In the frequency domain, however, the analysis can be further simplified. Figures 1(e) and (f) show the absolute values of the discrete Fourier transforms (DFT) of Figs. 1(a) and (b), respectively, after a logarithmic scaling for viewing purposes. Figure 1(f) shows very distinctive peaks at mid and high frequencies. Such peaks are due to the overlapping of pixel grids between the camera and the screen.

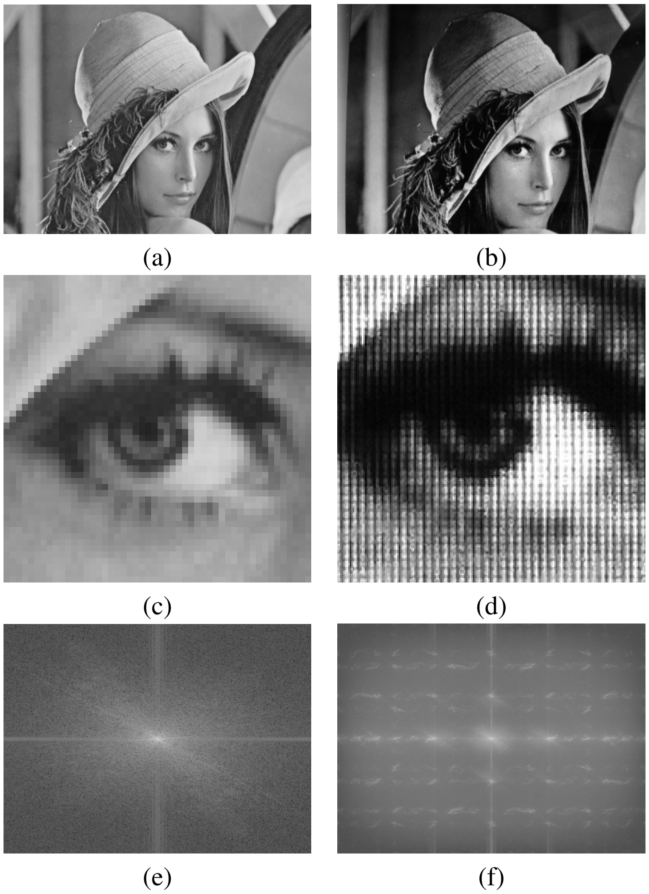


Fig. 1. Example of Moiré patterns due to the overlapping of digital grids. (a) Portion of the *Lena* test image. (b) Photograph of (a) on a 13-inch Macbook Pro screen and shot by an iPhone 4 camera without any compression artifacts. (c)-(d) Details of (a)-(b), respectively. (e)-(f) Absolute values of the discrete Fourier transforms of (a)-(b), respectively, after a logarithmic scaling for viewing purposes.

Moiré patterns have been thoroughly studied [22]-[24]. In order to simplify the analysis, let us look at the one-dimensional case. Consider a continuous-space low-pass function $f(t)$, which is to be sampled with period T_1, rendering $f_s(nT_1)$. The Fourier transforms of $f(t)$ and $f_s(nT_1)$ are $F(\omega)$ and $F_s(\omega) = \sum_k F(\omega - 2\pi k/T_1)$, respectively. When $f_s(nT_1)$ is displayed on screen, $f_{sd}(t)$ is rendered, which is equivalent to the convolution of $f_s(nT_1)$ with a boxcar function, or to the multiplication of $F_s(\omega)$ with a sinc function. Figure 2 illustrates this process.

$f_{sd}(t)$ is then recaptured by a digital camera with sampling interval T_2. In order for Moiré patterns to emerge after recapture, the sampling frequency $1/T_2$ on the digital camera must be larger than the screen sampling frequency $1/T_1$, or $T_1 > T_2$ [22]. Otherwise, the spectral repetitions shown in Fig. 2(f) at frequencies $1/T_1$, $2/T_1$ and so on will fall out of the frequency range of the recaptured image. It is important to point out that prior to resampling, low-pass filtering may take place, due to motion blur, lens defocus, diffraction and pixel response, among others, reducing the strength of the spectral repetitions and of the Moiré patterns.

The sampling interval T_2 depends on two factors: the size of the camera's pixels and the distance between the screen and the camera. As the distance from the camera to the screen increases, the sampling interval T_2 increases proportionally, reducing the capture resolution. The relation between T_1 and T_2 indicates if Moiré patterns are bound to occur, but it turns out to be very hard to directly measure. The sampling interval ratio $SR = T_1/T_2$, however, can be approximated by the pixel ratio PR:

$$PR = N_2/N_1, \tag{1}$$

where N_1 and N_2 represent the pixel lengths of a given feature on the screen and on the camera, respectively.

As the distance from the camera to the screen increases, T_2 also increases, decreasing SR. The pixel length N_2 of the given feature on the camera will decrease proportionally, and so will PR. Since it is necessary that $T_1 > T_2$ for the Moiré patterns to emerge, and assuming $SR \approx PR$, it immediately follows that:

$$PR > 1. \tag{2}$$

Figure 3 illustrates the dependency of the capture resolution on the distance from camera to screen.

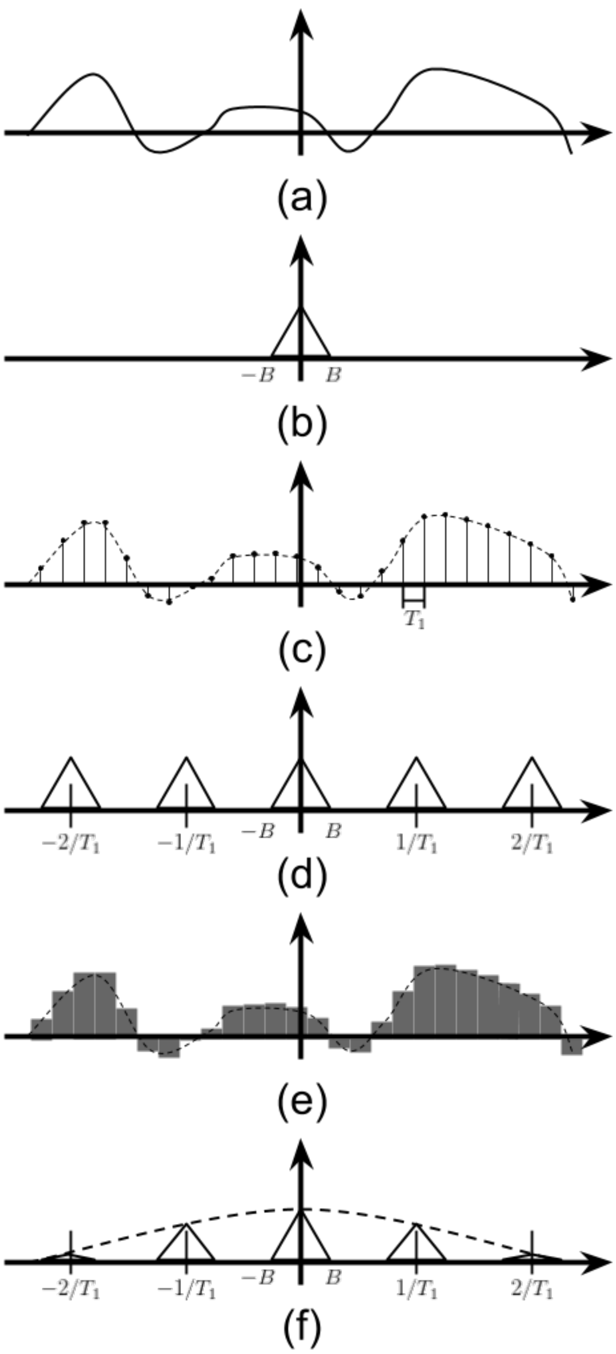


Fig. 2. Image display on screen (one-dimensional analysis): (a) continuous-space low-pass function $f(t)$ and (b) its Fourier transform $F(\omega)$; (c) sampling $f_s(nT_1)$ of $f(t)$ at regular intervals T_1 and (d) its Fourier transform $F_s(\omega)$; (e) $f_{sd}(t)$ representing $f_s(nT_1)$ displayed on a screen, which is equivalent to the convolution of $f_s(nT_1)$ with a boxcar function and (f) the Fourier transform $F_{sd}(\omega)$ of $f_{sd}(t)$, represented by the multiplication of $F_s(\omega)$ with a sinc function.

III. Proposed algorithm

The proposed algorithm assumes that regular, non-face-spoofing images contain most of their energy at the low frequencies, while face-spoofing images contain unusual peaks at higher frequencies.The goal is to find peaks at frequencies other than the baseband.

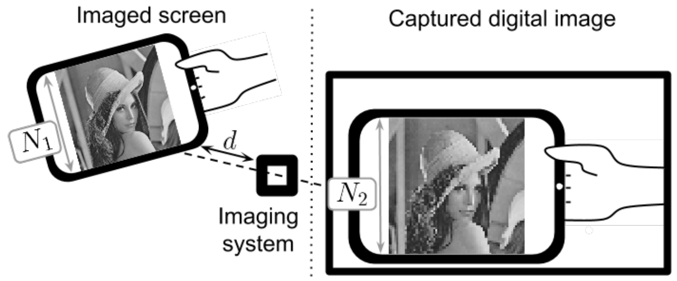


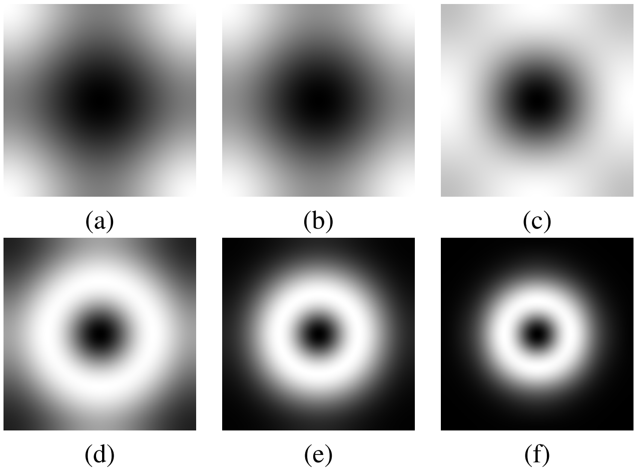
Fig. 3. Image recapture: a screened image is captured by an imaging system from a distance d. A given feature presents a length of N_1 pixels on the screen, and a length of N_2 pixels after recapture.

There is no easy way to determine *a priori* how much of the baseband needs to be ignored, so that the algorithm needs to search for peaks at different frequency bands. In essence, the algorithm works as follows. Given an image $\mathbf{I}$ of a detected face, distinct band-pass-filtered versions of this image are generated, and a peak detector is applied to the absolute value of the DFT of each of these filtered versions. If any strong peak is detected, the image is considered a face-spoofing image.

Each band-pass-filtered version of $\mathbf{I}$ ($\mathbf{I}_{BP}$) is obtained through convolution of $\mathbf{I}$ with a difference-of-Gaussians (DoG) filter [25]:

$$\mathbf{D}(\sigma, k) = \mathbf{G}(0, \sigma^2) - \mathbf{G}(0, k\sigma^2), \tag{3}$$

where $\mathbf{G}(0, \sigma^2)$ is a 2D-Gaussian function with zero mean and standard deviation σ. The argument k determines the width of the frequency band, and σ defines the center of the frequency band. In the proposed algorithm, several DoG filters are tested, with σ varying from σ_0 to σ_{max} in increments of Δ. Figures 4(a)-(f) illustrate the frequency response of DoG filters for $k = 2$, $\sigma_0 = 0.1$, $\sigma_{max} = 1.1$ and $\Delta = 0.2$.


(a) (b) (c)

(d) (e) (f)

Fig. 4. Absolute values of the DFT of DoG filters for $k = 2$: (a) $\sigma = 0.1$; (b) $\sigma = 0.3$; (c) $\sigma = 0.5$; (d) $\sigma = 0.7$; (c) $\sigma = 0.9$; (d) $\sigma = 1.1$.

Band-pass filtering of $\mathbf{I}$ is followed by peak detection in the frequency domain. The peak-detector algorithm is based on maximum-correlation thresholding [26], which works as follows: given any image $\mathbf{A}$, its thresholded version $\mathbf{B} = \mathcal{T}\{\mathbf{A}\}$ is defined as

$$B(u,v) = \mathcal{T}\{A(u,v)\} = \begin{cases} 1, & A(u,v) > t \\ 0, & A(u,v) \leq t \end{cases}. \tag{4}$$

In this case, t is the value that maximizes the correlation $\rho_{\mathbf{AB}}$ between $\mathbf{A}$ and $\mathbf{B} = \mathcal{T}\{\mathbf{A}\}$, which is defined as:

$$\rho_{\mathbf{AB}} = \frac{\mathrm{E}_{\mathbf{AB}} - \mathrm{E}_{\mathbf{A}}\mathrm{E}_{\mathbf{B}}}{\sqrt{(\mathrm{E}_{\mathbf{AA}} - \mathrm{E}_{\mathbf{A}}^2)(\mathrm{E}_{\mathbf{BB}} - \mathrm{E}_{\mathbf{B}}^2)}}, \tag{5}$$

where

$$\begin{aligned} \mathrm{E}_{\mathbf{A}} &= \sum_{g=0}^{n} g\, p(g) \\ \mathrm{E}_{\mathbf{AA}} &= \sum_{g=0}^{n} g^2\, p(g) \\ \mathrm{E}_{\mathbf{B}} &= \sum_{g=0}^{t} \mu_0(t)\, p(g) + \sum_{g=t+1}^{n} \mu_1(t)\, p(g) \\ \mathrm{E}_{\mathbf{BB}} &= \sum_{g=0}^{t} \mu_0^2(t)\, p(g) + \sum_{g=t+1}^{n} \mu_1^2(t)\, p(g) \\ \mathrm{E}_{\mathbf{AB}} &= \sum_{g=0}^{t} g\, \mu_0(t)\, p(g) + \sum_{g=t+1}^{n} g\, \mu_1(t)\, p(g) \end{aligned} \quad . \tag{6}$$

Here, g is one of the $n+1$ possible grey values of $A(u,v)$ and $p(g)$ is the probability of the grey level g. $p(g)$ is approximated by the histogram of $A(u,v)$. $\mu_0(t)$ and $\mu_1(t)$ are the below- and above-threshold means, such that $\mu_0(t) = (\sum_{g=0}^{t} g \;\; p(g))/(\sum_{g=0}^{t} p(g))$ and $\mu_1(t) = (\sum_{g=t+1}^{n} g\, p(g))/(\sum_{g=t+1}^{n} p(g))$.

If there are peaks on $|\mathcal{F}\{\mathbf{I}_{BP}\}|$ (the absolute values of the DFT of any of the band-pass-filtered versions of $\mathbf{I}$), maximum-correlation thresholding of this image will emphasize those peaks, and very few of its pixels will have a higher value than the threshold t. If $|\mathcal{F}\{\mathbf{I}_{BP}\}|$ does not contain peaks, more of its pixels will have a higher value than t.

In this manner, the peak-detector algorithm consists in thresholding $|\mathcal{F}\{\mathbf{I}_{BP}\}|$ and counting the percentage p of pixels with a higher value than the threshold t:

$$p = \frac{1}{WL}\sum_{u=1}^{W}\sum_{v=1}^{L}\mathcal{T}\{|\mathcal{F}\{\mathbf{I}_{BP}\}|\}, \tag{7}$$

where W is the image's width and L its height. If $p < p_{min}$, $\mathbf{I}$ is considered a face-spoofing image. The value p_{min} is a simple percentage of the whole image, and it is supposed to be very small when peaks are present in $|\mathcal{F}\{\mathbf{I}_{BP}\}|$.

The algorithm is repeated for different values of σ, and if no peak is found for all band-pass versions of $\mathbf{I}$, it is considered a non-face-spoofing image. The proposed algorithm is summarized in Figure 5.

Figure 6 shows output samples of the proposed algorithm. Figures 6(a) and (b) show $|\mathcal{F}\{\mathbf{I}_{BP}\}|$ of the detected faces in Figs. 1(a) and (b), with $\sigma = 0.1$ and $k = 2$. Figures 6(c) and (d) show maximum-correlated thresholded versions of Figs. 6(a) and (b), respectively. It can be seen in Fig. 6(d) that distinctive peaks are emphasized by the algorithm.

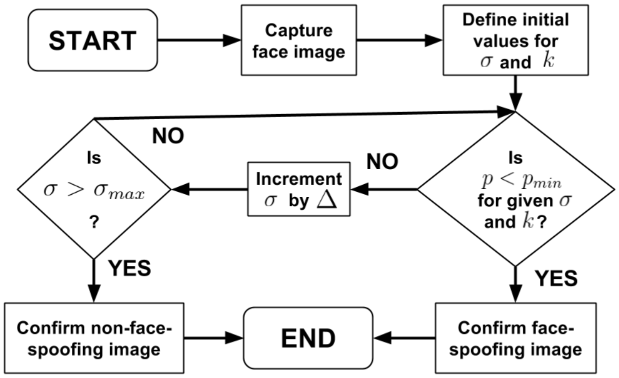

START
Capture face image
Define initial values for σ and k
Is $p < p_{min}$ for given σ and k?
NO
Increment σ by Δ
Is $\sigma > \sigma_{max}$?
NO
YES
YES
Confirm non-face-spoofing image
END
Confirm face-spoofing image

Fig. 5. Face-spoofing detection algorithm based on Moiré-pattern analysis.

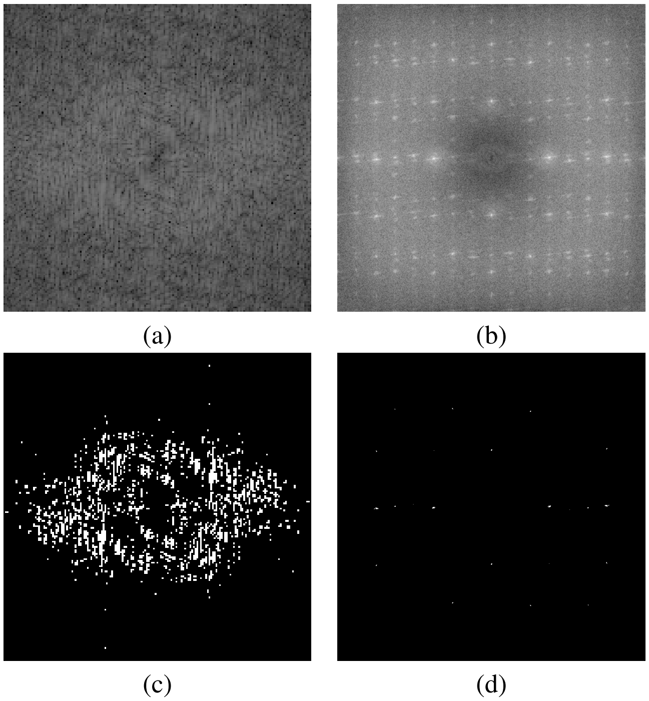


Fig. 6. Output samples of the proposed algorithm. (a)-(b) Absolute values of the DFT of the detected faces in Figs. 1(a)-(b) after convolution with a DoG filter, with $\sigma = 0.1$ and $k = 2$. (c)-(d) Maximum-correlated thresholded versions of (a)-(b), respectively. Logarithmic scaling was applied to (a)-(b) for viewing purposes.

IV. Experimental results

The proposed algorithm was validated with a large face-spoofing database, using images of 50 individuals under 13 conditions, in a total of 650 images. Face detection was performed using the OpenCV implementation of the Viola-Jones algorithm [30],[31]. In order to avoid border-continuity artifacts, a Hanning window was applied prior to the calculation of the DFTs, and the following parameters were empirically chosen: $\sigma_0 = 0.1$, $\sigma_{max} = 2.1$, $\Delta = 0.2$ and $p_{min} = 10^{-3}$. 9×9 kernels were used in the implementation of the DoG filters. The proposed algorithm was applied to the luminance component of the face-detected images.

Two separate experiments were made in order to account for the peculiarities of the many face-spoofing databases available online. Subsection IV-A presents the results with the Replay Attack Corpus, made available by the Idiap Research Institute, Martigny, Switzerland [12]. Subsection IV-B presents the results with a newly created face-spoofing database, the Moiré database, which better reflects the conditions described in Section II.

A. Experiments with the Replay Attack Corpus

Several authors make face-spoofing databases available, such as Zhang *et. al.* [3], Tan *et. al.* [4], Anjos and Marcel [7] and Chingovska *et. al.* [12]. These databases present faces of several individuals under many circumstances, resulting in hundreds of compressed video sequences. They also use low-resolution cameras, such as webcams. Compression artifacts and the lack of high resolution may hinder our algorithm ineffective, since we need to capture subtle Moiré patterns. That does not compromise our algorithm, it only reflects conditions that can be easily met by a real camera setup, that may not be present in building all databases.

To illustrate our concerns, we ran tests on the first 50 frames of four video sequences in the Replay-Attack Corpus. Figures. 7(a)-(d) present detected faces in the first frame of each of the used sequences. In a total of 200 frames, none of them were detected as attacks. For example, the DFT of Fig. 7(d) is presented in Fig. 7(e), which clearly does not present distinctive peaks as Figs. 1(f) and 6(b) nor yields distinctive secondary DFT peaks as in Fig. 7(f), thus making the database image capture conditions unsuitable to the proposed algorithm.

B. Experiments with the Moiré database

In order to generate face-spoofing images for our tests, a database comprised of 50 images of individuals under 13 different conditions was generated, with images from the MIT-CBCL Face Recognition Database, the Extended Yale Face Database B and the Frontal face dataset from the Computational Group at Caltech [27]-[29]. Even though the databases contain hundreds of images of the individuals under different poses and lighting directions, we fortuitously chose those with frontal pose and relatively frontal lighting, since the face-detector performance is not an issue in this work. Figures 8(a)-(c) present samples of the used databases.

The aforementioned images were photographed under several conditions: (i) displayed on a 13-inch Macbook Pro screen and captured by an iPhone 4 camera; (ii) displayed on a 13-inch Macbook Pro screen and captured by an iPad Mini camera; (iii) displayed on an iPad Mini screen and captured by an iPhone 4 camera; and (iv) displayed on an iPhone 4 screen and captured by an iPad Mini camera. Figures 8(d)-(f) present samples of Figs. 8(a)-(c) under conditions (ii)-(iv), respectively.

In all of these conditions, images were taken at different distances from the displays, in order to evaluate how the proposed algorithm operates under these circumstances. These different conditions were chosen so that two fundamental factors could separately be accounted for: (a) the distance between the display and the camera; and (b) the pixel resolution. In all of the aforementioned conditions, images were captured as uncompressed TIFF files [32], so as to avoid affecting the results by lossy-compression artifacts. Table I summarizes all tested conditions. The full database can be found at http://image.unb.br/queiroz/moiredatabase.

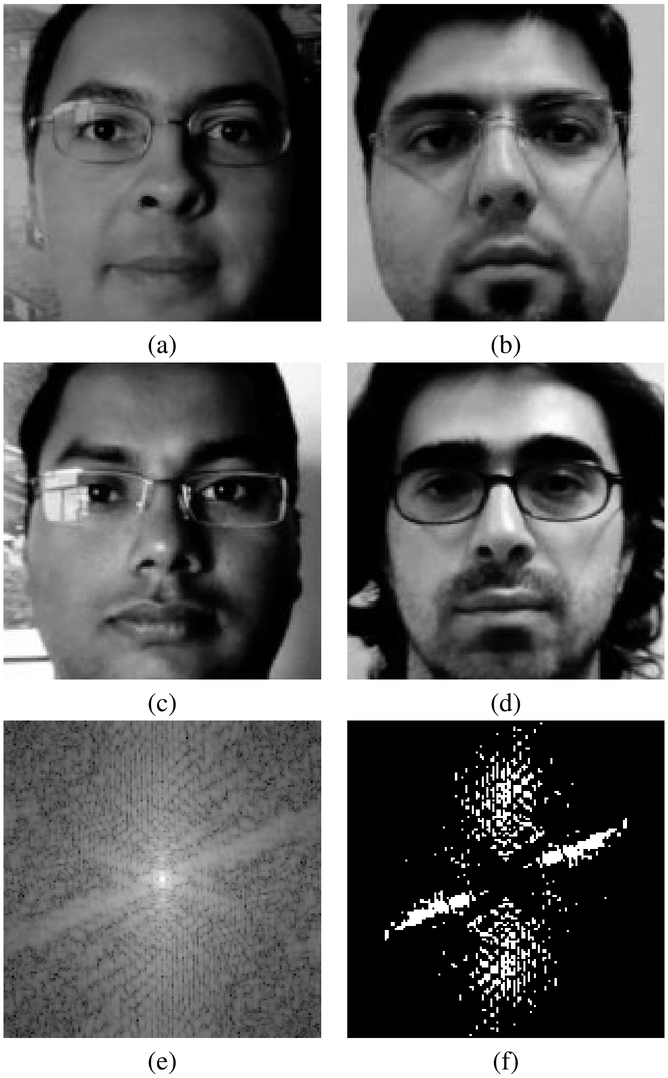

(a) (b) (c) (d) (e) (f)

Fig. 7. Algorithm output for some of the images in the Replay-Attack database [12]: (a)-(d) are, respectively, detected faces in the first frame of video sequences *attack_highdef_client004_session01_highdef_video_adverse.mov*, *attack_highdef_client006_session01_highdef_video_controlled.mov*, *attack_highdef_client001_session01_highdef_photo_adverse.mov* and *attack_highdef_client002_session01_highdef_photo_controlled.mov*, (e) is the absolute value of the DFT of (d), and (f) is the output of the algorithm for (d). (a)-(b) are found under the *train/attack/hand/* folder, and (c)-(d) are found under the *train/attack/fixed/* folder. Logarithmic scaling was applied to (e) for viewing purposes.

Figure 9 presents the false living rate under each condition in Table I as a function of the average pixel ratio PR. For each condition, the false living rate refers to the percentage of spoofing images that were not detected as such, and the average pixel ratio is measured as the average ratio of the widths of the detected faces on the capturing system (N_2) and on the face-spoofing screen (N_1). Condition 0 is not depicted, as the pixel ratio cannot be defined in this case. Under condition 0, no false positives were detected, yielding a null false spoofing, i.e. no non-spoofing image was detected

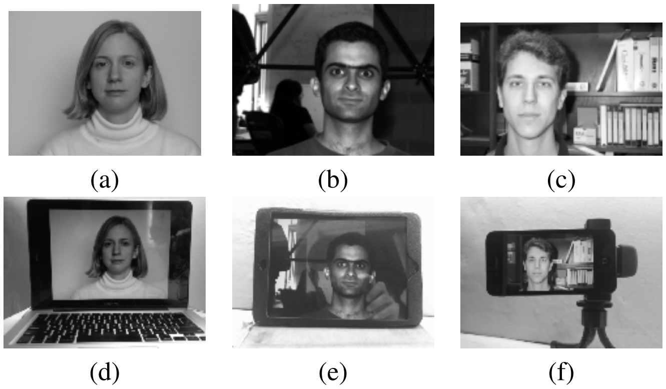


Fig. 8. Samples of the Moiré database, created to validate the proposed algorithm: (a)-(c) are, respectively, in the MIT-CBCL Face Recognition Database [27], the Extended Yale Face Database B [28] and the Frontal Face dataset from Caltech [29], and (d)-(f) are face-spoofing attacks for (a)-(c) under different conditions.

TABLE I
TESTED CONDITIONS FOR THE FACE-SPOOFING IMAGE DATABASE.

Condition	Display	Capture	Distance	Average PR
0	Original image			
1	Macbook Pro	iPhone 4	$\approx$ 20cm	3.08
2	Macbook Pro	iPhone 4	$\approx$ 30cm	1.93
3	Macbook Pro	iPhone 4	$\approx$ 40cm	1.40
4	Macbook Pro	iPad Mini	$\approx$ 20cm	2.97
5	Macbook Pro	iPad Mini	$\approx$ 30cm	1.99
6	Macbook Pro	iPad Mini	$\approx$ 40cm	1.46
7	iPad Mini	iPhone 4	$\approx$ 15cm	2.63
8	iPad Mini	iPhone 4	$\approx$ 20cm	1.91
9	iPad Mini	iPhone 4	$\approx$ 25cm	1.67
10	iPhone 4	iPad Mini	$\approx$ 10cm	1.86
11	iPhone 4	iPad Mini	$\approx$ 15cm	1.22
12	iPhone 4	iPad Mini	$\approx$ 20cm	0.92

as a spoof.

It can be seen from Fig. 9 that the condition in Eq. 2 holds true, and that the algorithm becomes more reliable as the pixel ratio increases. For $PR > 2$, no false negatives are reported. This behaviour is expected for the algorithm, since the increase in the pixel ratio reduces the effect of low-pass filtering due to motion blur, lens defocus, diffraction and pixel response, among others. For $1 < PR < 2$ however, a perfect score cannot be guaranteed for the algorithm.

This result suggests that the proposed algorithm requires a maximum distance between the screen and the capturing system in order to detect face-spoofing images. This can be achieved by requiring a minimum size for the detected face, so that an image in a tablet, laptop or smartphone would need to be close enough to the capturing system for the Moiré patterns to emerge. Also, the proposed algorithm requires a minimum resolution for the capturing system, so that low-resolution cameras such as cheaper webcams may not be adequate for the task at hand.

In order to clarify these remarks, Fig. 10 presents further details of the results of the proposed algorithm. Figures 10(a), (d) and (g) represent cases of a true negative, a true positive and a false negative, respectively. Figures 10(b), (e) and (h) present the absolute values of the DFT of Figs. 10(a), (d) and (g), respectively, after convolution with DoG filters. Figures 10(c), (f) and (i) present the thresholded versions of Figs. 10(b), (e) and (h), respectively.

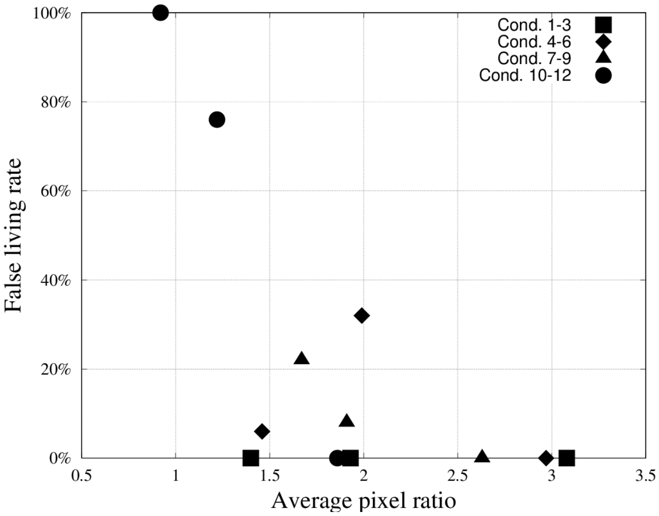


Fig. 9. False living rate for the proposed algorithm as a function of the average pixel ratio. The data was computed under conditions 1-12 in Table I. Condition 0 represents non-face-spoofing images, for which the false spoof rate was found null.

It can be seen from Fig. 10 that the algorithm is effective as long as the peaks of interest in the DFT are sufficiently prominent, such as in Figure 10(e). At large distances from the display to the camera and for low-resolution screens, these peaks do not appear, and the algorithm is not able to correctly detect face-spoofing images. This is the case for Figure 10(h), which was obtained under condition 12, an iPad Mini photographing an iPhone 4 at a 20 cm distance.

We also evaluated the proposed algorithm's performance as opposed to Zhang *et. al.*'s algorithm [3], which involves the training and testing of support vector machines. Since the Moiré database contains 50 non-spoofing images and $50 \times 12 = 600$ spoofing images, it represents a very unbalanced dataset for training and testing, yielding unsatisfactory results [20]. The best solution found was to train and test separate SVMs for each condition. For instance, we chose 25 images from Condition 0 and 25 images from Condition 1 to train the SVM, which was tested on the remaining 25 images from each of these Conditions. We repeated this for Conditions 0 and 2, 0 and 3 and so on. We used the LIBSVM library, employed a radial basis function (RBF) kernel, and chose the best results from a wide range of C and γ values [33].

Figures 11 and 12 present the false living rate and false spoofing rate, respectively, for Zhang *et. al.*'s algorithm, compared to ours. Results indicate that, in terms of false living rate, the proposed algorithm is outperformed for low pixel ratio values ($PR \approx 1$), but Zhang *et. al.*'s algorithm cannot guarantee a null false living rate without sacrificing the false spoofing rate. As shown in Fig. 12, Zhang *et. al.*'s algorithm also does not guarantee a null false spoofing rate, as opposed to the proposed algorithm. The results shown in Figs. 11 and 12 may not be fully representative of the method, given the reduced number of training images, but they also show that spoofing detection based on empirical classification, such as SVMs and neural nets, can be highly dependent on the training dataset, making it much more difficult to implement in real-world situations.

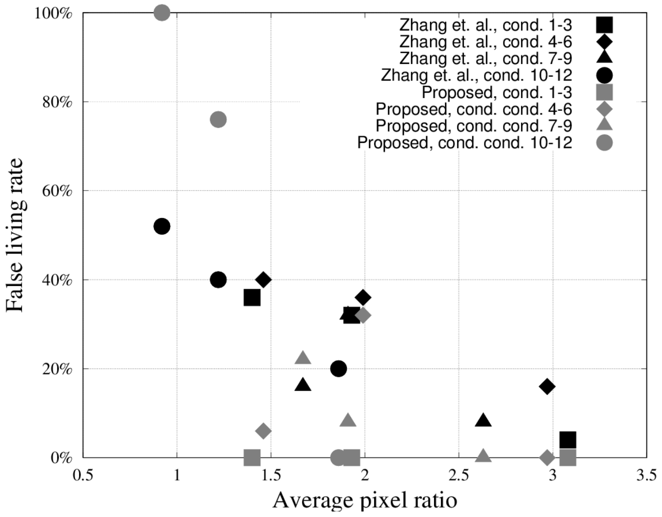


Fig. 11. False living rate for Zhang *et. al.*'s algorithm [3] as a function of the average pixel ratio. The data was computed under conditions 0-12 in Table I. The results for the proposed algorithm are also shown for comparison.

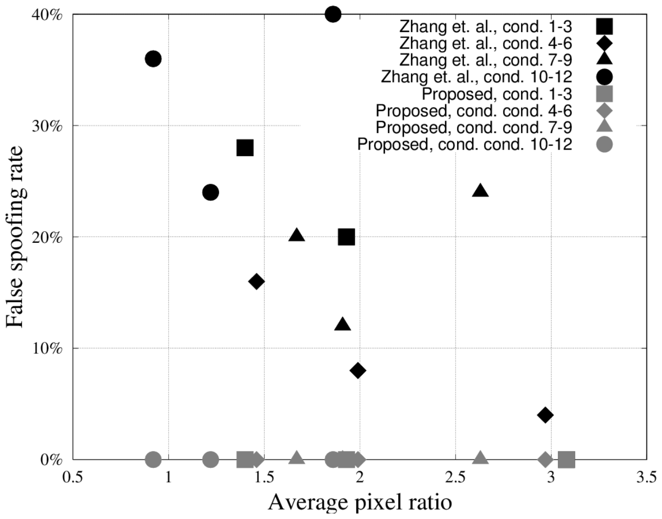


Fig. 12. False spoofing rate for Zhang *et. al.*'s algorithm [3] as a function of the average pixel ratio. The data was computed under conditions 0-12 in Table I. The results for the proposed algorithm are also shown for comparison.

V. Conclusions

In this paper, a face-spoofing detection algorithm was proposed based on the detection of Moiré patterns due to the overlap of digital grids. The conditions under which these patterns arise were described and experimentally verified. The effectiveness of the proposed algorithm was also verified by running tests on a database of face images shot under several conditions. Results show that, under the right conditions, face-spoofing detection can be performed with great accuracy.

References

[1] S. Prabhakar, S. Pankanti, and A. K. Jain, "Biometric recognition: Security and privacy concerns," *IEEE Security Privacy*, vol. 1, no. 2, pp. 3342, Mar./Apr. 2003.

[2] J. Li, Y. Wang T. Tan and A. Jain, "Live face detection based on the analysis of Fourier spectra," *Biometric Technology for Human Identification*, pp. 296–303, 2004.

[3] Z. Zhang, J. Yan, S. Liu, Z. Lei; D. Yi, S. Z. Li, "A face antispoofing database with diverse attacks," *IEEE Int. Conf. on Biometrics Compendium*, pp.26–31, Mar. 2012.

[4] X. Tan, Y. Li, J. Liu and L. Jiang, "Face liveness detection from a single image with sparse low rank bilinear discriminative model," *11th European Conf. on Computer Vision*, pp 504-517, Sep. 2010.

[5] B. Peixoto, C. Michelassi and A. Rocha, "Face liveness detection under bad illumination conditions," *IEEE Int. Conf. Image Process.*, pp. 3557,3560, Sept. 2011.

[6] J. Galbally, S. Marcel, and J. Fierrez, "Image quality assessment for fake biometric detection: application to iris, fingerprint, and face recognition," *IEEE Trans. Image Process.*, vol.23, no.2, pp.710–724, Feb. 2014.

[7] A. Anjos and S. Marcel, "Counter-measures to photo attacks in face recognition: a public database and a baseline," *Int. Joint Conf. on Biometrics*, pp.1–7, Oct. 2011.

[8] M. De Marsico, M. Nappi, D. Riccio and J. Dugelay, "Moving face spoofing detection via 3D projective invariants," *IAPR Int. Conf. Biometrics*, pp.73–78, Mar. 2012.

[9] K. Kollreider, H. Fronthaler and J. Bigun, "Non-intrusive liveness detection by face images," *Image and Vision Computing*, vol. 27, issue 3, pp. 233-244, Feb. 2009.

[10] A. Pinto, H. Pedrini, W. Schwartz and A. Rocha, "Video-based face spoofing detection through visual rhythm analysis," *SIBGRAPI Conf. Graphics, Patterns and Images*, pp.221–228, Aug. 2012.

[11] J. Määttä, A. Hadid and M. Pietikäinen, "Face spoofing detection from single images using micro-texture analysis," *Int. Joint Conf. on Biometrics*, pp.1–7, Oct. 2011.

[12] I. Chingovska, A. Anjos, S. Marcel, "On the effectiveness of local binary patterns in face anti-spoofing," *Proc. Int. Conf. Biometrics Special Interest Group*, pp.1–7, Sept. 2012.

[13] S. Bharadwaj, T.I. Dhamecha, M. Vatsa and R. Singh, "Computationally efficient face spoofing detection with motion magnification," *IEEE Conf. Comput. Vision and Pattern Recognition Workshops*, pp.105–110, June 2013.

[14] H.-Y. Wu, M. Rubinstein, E. Shih, J. Guttag, F. Durand and W. T. Freeman, "Eulerian video magnification for revealing subtle changes in the world", *ACM Transactions on Graphics*, 31(4), 2012.

[15] G. Kim, S. Eum, J. Suhr, D. Kim, K. Park and J. Kim, "Face liveness detection based on texture and frequency analyses," *IEEE Int. Conf. on Biometrics Compendium*, pp.67–72, Mar. 2012.

[16] T. Pereira, J. Komulainen, A. Anjos, J.M. De Martino, A. Hadid, M. Pietikäinen and S. Marcel, "Face liveness detection using dynamic texture.," *EURASIP J. Image and Video Process.*, vol. 2014, no. 1, pp. 1–15, 2014.

[17] J. Bai, T.-T. Ng, X. Gao and Y.-Q. Shi, "Is physics-based liveness detection truly possible with a single image?," *Proc. IEEE Int. Symp. Circuits Syst.*, pp. 3425–3428, May 2010.

[18] W.R. Schwartz, A. Rocha and H. Pedrini, "Face spoofing detection through partial least squares and low-level descriptors," *IEEE Int. Conf. on Biometrics Compendium*, pp.1–8, Oct. 2011.

[19] T. Pereira, A. Anjos, J. M. De Martino and S. Marcel, "Can face anti-spoofing countermeasures work in a real world scenario?," *IEEE Int. Conf. on Biometrics Compendium*, pp.1–8, June 2013.

[20] R. Akbani, S. Kwek and N. Japkowicz, "Applying support vector machines to imbalanced datasets," *Proc. European Conf. Machine Learning*, pp. 39–50, 2004.

[21] I. Amidror, "The theory of the Moiré phenomenon, volume I: periodic layers," Springer, 2nd edition, 2009.

[22] J. Krumm and S. Shafer, "Sampled-grating and crossed-grating models of Moiré patterns form digital imaging," *Optical Engineering*, vol. 30, no. 2, pp. 195–206, 1991.

[23] J. Allebach and B. Liu, "Analysis of halftone dot profile and aliasing in the discrete binary representation of images," *J. Opt. Soc. Am.*, vol. 67, pp. 1147–1154, 1977.

[24] A. Steinbach and K. Wong, "Moiré patterns in scanned halftone pictures," *J. Opt. Soc. Am.*, vol. 72, pp. 1190–1198, 1982.

[25] J. Crowley and R. Stern, "Fast computation of the difference of low-pass transform," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol.PAMI-6, no.2, pp.212–222, March 1984.

[26] A. Brink, "Grey–level thresholding of images using a correlation criterion," *Pattern Recognition Lett.*, vol. 9, issue 5, pp. 335–341, June 1989.

[27] B. Weyrauch, B. Heisele, J. Huang and V. Blanz, "Component-based face recognition with 3D morphable models," *Proc. of CVPR Workshop on Face Processing in Video*, June 2004.

[28] K. Lee, J. Ho and D. Kriegman, "Acquiring linear subspaces for face recognition under variable lighting," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 5, pp. 684–698, May 2005.

[29] Frontal face dataset from the Computational Group at Caltech, *http://vision.caltech.edu/archive.html*.

[30] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," *Proc. of CVPR*, vol.1, pp. I-511–I-518, 2001.

[31] G. Bradski, "The OpenCV Library," *Dr. Dobb's Journal of Software Tools*, 2000.

[32] 645 Pro Mk II iOS app, *http://jag.gr/645pro/*.

[33] C.-C. Chang and C.-J. Lin, "LIBSVM: a library for support vector machines," *ACM Trans. Intell. Syst. Tech*, 2:27:1–27:27, 2011.

Diogo C. Garcia received the Electrical Engineer, M.Sc. and Ph.D. degrees in electrical engineering from the University of Brasilia, Brasilia, Brazil, in 2006, 2008 and 2012, respectively.

He is currently an Adjunct Professor with the Gama Faculty of the University of Brasilia. His current research interests include image and video coding, super resolution, and multiview and 3-D processing.

Ricardo L. de Queiroz received the Engineer degree from Universidade de Brasilia, Brazil, in 1987, the M.Sc. degree from Universidade Estadual de Campinas, Brazil, in 1990, and the Ph.D. degree from The University of Texas at Arlington, in 1994, all in Electrical Engineering.

In 1990-1991, he was with the DSP research group at Universidade de Brasilia, as a research associate. He joined Xerox Corp. in 1994, where he was a member of the research staff until 2002. In 2000-2001 he was also an Adjunct Faculty at the Rochester Institute of Technology. He joined the Electrical Engineering Department at Universidade de Brasilia in 2003. In 2010, he became a Full Professor at the Computer Science Department at Universidade de Brasilia. He is now a Visiting Professor at the University of Washington, in Seattle.

Dr. de Queiroz has published over 160 articles in Journals and conferences and contributed chapters to books as well. He also holds 46 issued patents. He is a past elected member of the IEEE Signal Processing Society's Multimedia Signal Processing (MMSP) and the Image, Video and Multidimensional Signal Processing (IVMSP) Technical Committees. He is a an editor for IEEE Transactions on Image Processing and a past editor for the EURASIP Journal on Image and Video Processing, IEEE Signal Processing Letters, and IEEE Transactions on Circuits and Systems for Video Technology. He has been appointed an IEEE Signal Processing Society Distinguished Lecturer for the 2011-2012 term.

Dr. de Queiroz has been actively involved with the Rochester chapter of the IEEE Signal Processing Society, where he served as Chair and organized the Western New York Image Processing Workshop since its inception until 2001. He is now helping organizing IEEE SPS Chapters in Brazil and just founded the Brasilia IEEE SPS Chapter. He was the General Chair of ISCAS'2011, and MMSP'2009, and is the General Chair of SBrT'2012. He was also part of the organizing committee of ICIP'2002, ICIP'2012, ICIP'2014 and ICIP'2016. His research interests include image and video compression, multirate signal processing, and color imaging. Dr. de Queiroz is a Senior Member of IEEE, a member of the Brazilian Telecommunications Society and of the Brazilian Society of Television Engineers.

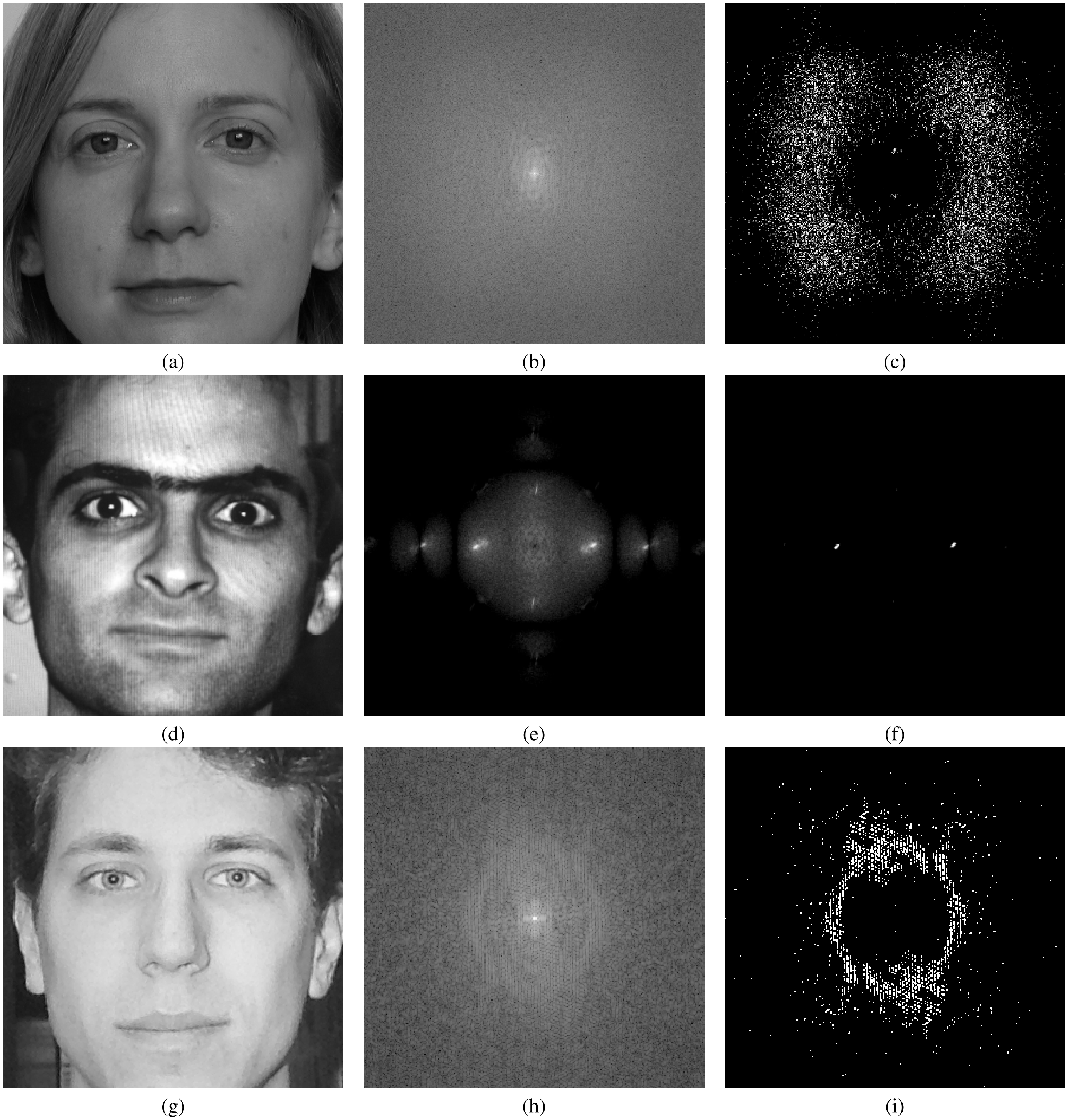


Fig. 10. True negative, true positive and false negative examples: (a) detected face in Figure 8(a); (b) natural logarithm of the absolute values of the DFT of (a); (c) Output of the algorithm for $\sigma = 0.1$, yielding a true negative; (d) detected face $\mathbf{I}$ for Figure 8(b) under Condition 1; (e) natural logarithm of the absolute values of the DFT of $\mathbf{I} * \mathbf{D}(0.1)$; (f) Output of the algorithm, yielding a true positive; (g) detected face for Figure 8(e) under Condition 12; (h) natural logarithm of the absolute values of the DFT of (a); (i) Output of the algorithm for $\sigma = 0.1$, yielding a false negative. Images are best seen on a screen.