Available online at www.sciencedirect.com
ScienceDirect
Image and Vision Computing 27 (2009) 233–244
image and vision COMPUTING
www.elsevier.com/locate/imavis

Non-intrusive liveness detection by face images

K. Kollreider *, H. Fronthaler, J. Bigun

Halmstad University, SE-30118, Sweden

Received 18 February 2006; received in revised form 24 January 2007; accepted 22 May 2007

Abstract

A technique evaluating liveness in face image sequences is presented. To ensure the actual presence of a live face in contrast to a photograph (playback attack), is a significant problem in face authentication to the extent that anti-spoofing measures are highly desirable. The purpose of the proposed system is to assist in a biometric authentication framework, by adding liveness awareness in a non-intrusive manner. Analyzing the trajectories of certain parts of a live face reveals valuable information to discriminate it against a spoofed one. The proposed system uses a lightweight novel optical flow, which is especially applicable in face motion estimation based on the structure tensor and inputs of a few frames. For reliable face part detection, the system utilizes a model-based local Gabor decomposition and SVM experts, where selected points from a retinotopic grid are used to form regional face models. Also the estimated optical flow is exploited to detect a face part. The whole procedure, starting with three images as input and finishing in a liveness score, is executed in near real-time without special purpose hardware. Experimental results on the proposed system are presented on both a public database and spoofing attack simulations.

Keywords: Face liveness; Liveness detection; Anti-spoofing measures; Optical flow; Motion of lines; Optical flow of lines; Orientation estimation; Face part models; Retinotopic vision; Local Gabor decomposition; Support vector machine classification

1. Introduction

Liveness detection is a highly desirable, yet rather unexplored anti-spoofing measure in biometric identity authentication [1,2]. Especially in face analysis only a few approaches address this subject: in [3], a depth map is constructed by recovering 3D structure from motion, which is in idea similar to our approach, since a live head and a moved photograph generate different depth maps. To the contrast, motion is estimated by a correlation based method there, and no experimental results are given. Another way is to analyze the frequency spectrum of a live face [4], defining two descriptors to measure the high frequency proportion and the temporal variance of all frequencies. This method presupposes both a lack of quality of a photograph (low-resolution) and the change of mimics and poses in a live face. It is commonly agreed on that a single 2D face image acquired by a traditional camera is not sufficient for reliable liveness detection.

In our approach, we combine face part detection and optical flow estimation to determine a liveness score. The typical trajectory of certain face parts in case of a live face sequence is exploited to discriminate it against a spoofed one. We use the *optical flow of lines*, [5], (OFL) which is inspired by optical flow approaches capable to differ between motion of points and motion of lines (e.g. Tensor approach [6]). As the name suggests, it is specialized on motion of lines only. Requiring only 2–3 images, the OFL approach is a lightweight energy based OF method, which can be realized by using 2D Gabor filters [5,7–9]. However, in this study we implement the OFL by employing 1D Gaussians and their derivatives. A review of optical flow techniques is given in [10].

For face part detection we combine OF pattern matching with a model-based technique employing Gabor features on a log-polar grid [11,12] and SVM [13]. Gabor

* Corresponding author.
E-mail addresses: klaus.kollreider@ide.hh.se (K. Kollreider), hartwig.fronthaler@ide.hh.se (H. Fronthaler), josef.bigun@ide.hh.se (J. Bigun).

0262-8856/$ - see front matter © 2007 Elsevier B.V. All rights reserved.
doi:10.1016/j.imavis.2007.05.004

filters are a class of powerful face recognition features [12,14,15] which have impulse responses resembling those of simple cells in the visual cortex [16–18]. Jumping to assumed points of interest, [19], instead of an exhaustive search, is characteristic for biological vision systems by which our retinotopic vision approach is inspired. Following an economic strategy, only specific frequency and orientation channels are used for modelling a facial region. Also, recognition rates can be raised by reducing the amount and adapting the range of frequencies for the decomposition [13,20]. Since moving face parts are dealt with, a significant speed up for face part detection can be experienced by discarding unchanged areas.

This paper is structured as follows: in the next chapter, the basic strategy and the application field of our liveness detection will be introduced. Chapter 3 presents the OFL, whereas chapter 4 outlines the face part detection. Chapter 5 describes the liveness detection system. Experimental results are presented in chapter 6, whereas chapters 7 and 8 discuss and conclude the main findings of the paper, respectively.

2. Basic strategy

Regardless of their authentication performance, all biometric systems will suffer if they cannot distinguish between a photograph and the live presence of a client. A poorly investigated problem in face authentication studies is the claim of someone else's identity by using a high quality photograph, whether in motion or not. Essentially three possibilities to make such a system liveness aware can be identified:

(1) Deploying a multi-modal system, [13,21–24], with numerous sensors (e.g. several cameras including stereo, heat sensitive cameras, etc).
(2) Interacting with the client (e.g. Automated Teller Machine) demanding real-time responses (e.g. talk, blink, etc).
(3) Exploiting the motion characteristics of a 3D face by using an image sequence.

Being the least studied, we will dwell on the third alternative in this paper. Indeed, the first two approaches do not compete with the third but complement it well. Our method analyzes a face image sequence captured by one camera and delivers a probability, whether it detected a face and whether it is live.

The basic idea relies on the assumption that a 3D face generates a special 2D motion which is higher at central face parts (e.g. nose) compared to the outer face regions (e.g. ears). Ideally, in terms of liveness detection, the outer and the inner parts move additionally in opposite directions. This case is visualized in Figs. 1 and 2, respectively, where a head slightly rotates to the left (from the person's view). Fig. 2 shows the horizontal OFL (optical flow estimate in horizontal direction only) from the image sequence displayed in Fig. 1. The rectangles indicate the focused face parts and their motion (note the signs).

In other words, parts nearer to the camera move differently to parts which are further away in a live face. The experimental results, we report below, support that a small rotational movement of the head is natural and unintentional human behavior. On the contrary, a translated photograph generates constant motion at various face regions. In order to exploit these characteristics, we utilize optical flow estimation and face part detection. For the latter we employ a model-based Gabor decomposition [13], but we also present an intuitive approach by OF pattern matching. Knowing the face parts' position and comparing how fast they are moving relative to each other and into which directions, enables us to discriminate a live face against a photograph.

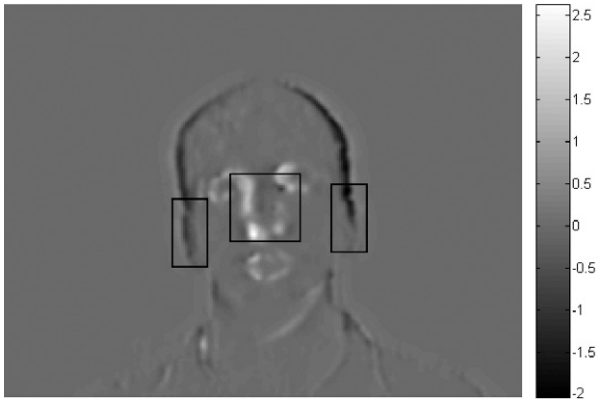

Fig. 2. Horizontal OFL (Gabor-based [5]) at the center frame of the image sequence in Fig. 1. Rectangles indicate the focused regions when comparing different face parts' motion.

Fig. 1. Example face image sequence: differently moving center and side face parts suggest the "live" presence of a person.

3. Optical flow estimation

The subsequently presented optical flow of lines (OFL) relies on some assumptions and simplifications. First, our OFL method can only handle motion of lines, referred to as normal motion. Second, it assumes lines to be either horizontal or vertical when estimating the velocity components. We motivate these simplifications by regarding lines, and especially those horizontally and vertically oriented, as the dominant structures in a face image of a known scale range. We assume these features to be sufficiently robust for spatiotemporal analysis. This allows, as will be detailed below, to reduce a 3D-minimization problem to 2D. Furthermore, in contrast to other optical flow methods, the OFL is computationally lightweight, which is not only due to its normal velocity restriction but also a consequence of requiring only 3 time frames.

3.1. Theoretical approach

In the general case of parallel lines undergoing a constant motion, parallel planes are generated in the spatiotemporal space, having a common normal unit vector $\hat{k} = (k_x, k_y, k_t)$ to describe them. The tilt of this normal vector with regard to the xy-plane corresponds to the absolute velocity of the concerned lines in 2D, which is also stated in the following equation (Fig. 3):

$$|v| = \tan\alpha = \frac{|k_t|}{\sqrt{k_x^2 + k_y^2}}. \tag{1}$$

In Figs. 3 and 4, the planes (gray) represent a single moving line. Due to the aperture problem, we can only determine the normal optical flow, which is in the spatial direction of $\hat{k}$. The horizontal and vertical velocity components are denoted in the following equations:

$$v_x = \cos\beta \cdot (-\tan\alpha) = \frac{k_x}{\sqrt{k_x^2 + k_y^2}} \cdot \frac{-k_t}{\sqrt{k_x^2 + k_y^2}} = -\frac{k_x \cdot k_t}{k_x^2 + k_y^2} \tag{2}$$

$$v_y = \cos\gamma \cdot (-\tan\alpha) = \frac{k_y}{\sqrt{k_x^2 + k_y^2}} \cdot \frac{-k_t}{\sqrt{k_x^2 + k_y^2}} = -\frac{k_y \cdot k_t}{k_x^2 + k_y^2}. \tag{3}$$

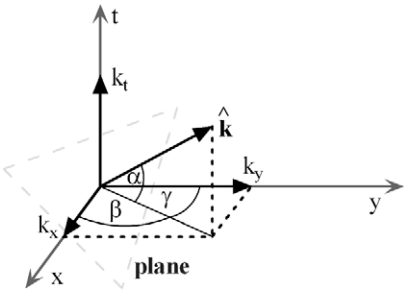

Fig. 3. Normal velocity in the spatiotemporal space: the plane represents a spatial linear symmetry tracked through time, i.e. moving straight edges as compared to moving points.

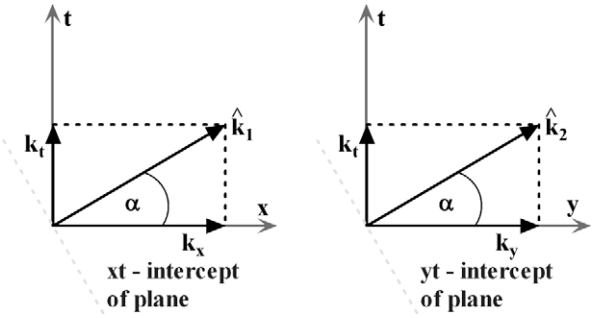

Fig. 4. Normal velocity in 2D xt- and yt-images: the plane in Fig. 3 is considered parallel to the y- and x-axis, respectively.

As a consequence, we need to estimate $\hat{k}$ in order to calculate the normal velocity. It has been shown that orientation estimation in 3D (and in higher dimensions) can be achieved by fitting a line or a plane to the local Fourier transform. This is further equivalent to an eigenvalue analysis of the multidimensional structure tensor, constructed by the averaged auto outer product of the spatiotemporal gradient. This allows the minimization process of fitting a line or a plane to be carried out without actually Fourier transforming [6]. Applied to optical flow estimation, known as tensor method, the eigenvector belonging to the solely large eigenvalue of the 3D structure tensor directs into the direction of $\hat{k}$, if normal motion is detected (spectral energy is concentrated to a line). Though easy to implement, the method requires many time frames and is computationally exhaustive which has been our main motivation to investigate a "light-weight" version of it.

If we have vertical and horizontal lines separately throughout the image sequence (achieved by directional filtering), we can formulate the normal optical flow in a spatially separable manner. We are then looking for tilted planes, which stay parallel to either of the spatial axes, y or x. The determination of these tilts corresponds to a 2D orientation estimation around the parallel axis, which is in xt- and yt-dimensions. Fig. 4 shows such images and the 2D normal vectors $\hat{k}_1$ and $\hat{k}_2$. Fig. 3 is related to Fig. 4 and the 2D directions by considering, respectively, k_y and k_x zero, the equivalent of setting β and γ to zero. This is also valid for the horizontal and vertical velocity components, which reduce to:

$$v_x = -\tan\alpha = -\frac{k_t}{|k_x|} \tag{4}$$

$$v_y = -\tan\alpha = -\frac{k_t}{|k_y|}. \tag{5}$$

The optical flow estimation as stated in Eq. (6) is thus determined by orientation in two dimensions for each component.

$$v = \begin{bmatrix} v_x \\ v_y \end{bmatrix} = \begin{bmatrix} -\frac{k_t}{|k_x|} \\ -\frac{k_t}{|k_y|} \end{bmatrix}. \tag{6}$$

In 2D orientation estimation can eigenvalue analysis of the corresponding 2D structure tensor be replaced by averaging both the square of a complex valued gradient image and its absolute value [6]. If we denote fx and fy as the image sequences containing extracted vertical and horizontal lines, respectively, we can establish the relationships as in the following equations.

$$\hat{k}_1^2 = (k_x + \mathrm{i}\cdot k_t)^2 = \int\int \left(\frac{\partial fx}{\partial x} + \mathrm{i}\cdot\frac{\partial fx}{\partial t}\right)^2 \mathrm{d}x\,\mathrm{d}t = V_x \qquad (7)$$

$$\hat{k}_2^2 = (k_y + \mathrm{i}\cdot k_t)^2 = \int\int \left(\frac{\partial fy}{\partial y} + \mathrm{i}\cdot\frac{\partial fy}{\partial t}\right)^2 \mathrm{d}y\,\mathrm{d}t = V_y. \qquad (8)$$

This corresponds to linear symmetry detection in xt- and yt-images, respectively. The complex numbers V_x and V_y directly encode the optimal direction in double angle representation, [25], and the error. In other words, $\arg(V_x)$ and $\arg(V_y)$ equal 2α in Eqs. (4) and (5), respectively, yielding the estimated velocity components stated in the following equations:

$$v_x = -\tan\left(\frac{1}{2}\arg(V_x)\right) \qquad (9)$$

$$v_y = -\tan\left(\frac{1}{2}\arg(V_y)\right). \qquad (10)$$

3.2. Implementation

In what follows are the input images referred to as Im_l, with $l = 1,2,3$. First, a region-of-interest image roi is derived by setting a threshold to the intensity differences between each of two subsequent images. A cube is cut out of the space-time stack with its spatial dimensions arranged to contain the relevant portion (motion) obtained by simple analysis of roi. In the following, we show an OFL implementation utilizing 1D Gaussians and their first derivatives, essentially for orientation estimation [26]. As to be seen, some advantages over the Gabor-based method suggested in [5], can be experienced.

First, we summarize the calculation of, for example, v_x. The vertical lines are approximatively extracted in each Im_l by applying a Gaussian column and a derivative Gaussian row filter sequentially. Then are xt-slices taken along the y-axis, and for each one is Eq. (7) evaluated in two steps: first, $\langle (x+it)\cdot g, |xt|\rangle = \langle g(t), \langle x\cdot g(x), |xt|\ \rangle\rangle + \mathrm{i}\langle g(x), \langle t\cdot g(t), |xt|\rangle\rangle$, where g is denoting the Gaussian function with σ_1 is calculated. Second, the result is averaged using another, larger Gaussian ($\sigma_2 > \sigma_1$) in x- and t-direction, respectively, to get V_x, which is also indicated by the double integral in Eq. (7). The value of v_x is obtained by applying Eq. (9). In a further step, two constraints are employed to assure that the estimated velocity at site (x,y) is reliable: (i) the maximal velocity is $|v_x(x,y)| < \tau_1$ and (ii) significant line structure must be present, $|\langle g(y), \langle x\cdot g(x), \mathrm{Im}_2(x,y)\rangle\rangle| > \tau_2$, which can be reused from the first step (extraction of line structure). The calculation of v_y is analogous, but the space-time stack is rotated by 90 deg around the t-axis.

Due to a tiny dimension of three pixels in t-direction σ_1, σ_2 need to be chosen small in order to avoid violating the design rules. Furthermore, the filters are likely to become anisotropic due to sampling errors. We suggest inserting a blank frame, before and after Im_2 into the space-time stack to artificially stretch the t-dimension to five pixels. After this modification, the results for v_x are significantly better, although we are introducing some aliasing due to not interpolating the blank images. Note that interlacing the space-time stack into t-dimension leads to a systematic error in the velocity estimation, which we have to correct. If the "interlaced" and the original orientation angle are denoted by β and α, respectively, then their relationship is given by $\frac{5}{3}\cdot\tan(\beta) = \tan(\alpha)$ using geometry. This means that we have to take v_x and v_y as in Eq. (9) but amplify it with 5/3 to correct the results. When using a bigger number of original images (e.g. 5) for the OFL, we do not need to interlace.

We combine the horizontal and vertical component velocities to a complex image $\mathrm{OF}_{\mathrm{im}}$ having v_x in its real part and v_y in the imaginary part as described in the following equation:

$$\underset{x,y\in \mathrm{Im}_2}{\mathrm{OF}_{\mathrm{im}}} = v_x(x,y) + \mathrm{i}\cdot v_y(x,y). \qquad (11)$$

The proposed implementation of the OFL together with the one described in [5] (Gabor-based) are applied on two test sequences, (a) a sinusoid image sequence undergoing a diagonal top to bottom translation by 2 pixels/picture, and, (b) a circle image sequence exposed to a vertical top to bottom movement by 2 pixels/picture. The results for the Gabor-based and Gauss-based method are shown on the left- and right-hand side of Fig. 5, respectively. In all images, the flow arrows taken from $\mathrm{OF}_{\mathrm{im}}$ are superimpos-

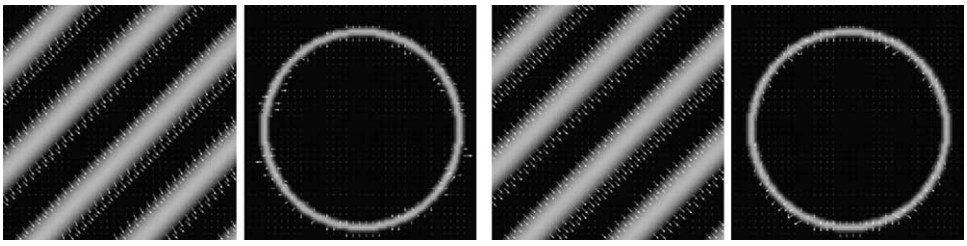

Fig. 5. Center frames of two test sequences: the OFL is calculated employing both Gabor-based (left) [5] and Gauss-based (right) methods for each of the sequences.

ing the center frame of the respective image sequence. As can be observed, by considering the horizontal and vertical portions of off-axis lines, the OFL is accurate at perfectly horizontal and vertical lines and reasonably accurate at oblique directions. The Gauss-based method is granted better accuracy by inspection, which can be explained by its ability to measure orientation continuously, while the Gabor-based method is essentially equipped with three selective orientations and is only approximating orientations in between [27]. Computational load is another important point concerning performance of the two implementation approaches. Given similar Gabor and Gaussian filter sizes and the discussed space-time stack depth, the efforts for calculating the component velocities at each pixel of the center image are approximately the same for both methods. This was also confirmed empirically and could be shown simply by counting the operations but is omitted here. This is, however, only possible due to the small filter sizes because the number of operations are fewer for larger filters considering the separability of the Gaussian-based filters. Also, it is subject to specific implementation and used hardware: separable filtering is often optimized in common computer architectures which will favor the Gauss-based method again.

4. Face part detection

We wish to track the three facial regions eyes/nose, left and right ear. To reliably detect these face parts we combine optical flow pattern matching and a model-based Gabor feature classification. The assumption of having a moving face is exploited to speed up its parts' detection by taking the region-of-interest image roi into account.

4.1. Optical flow pattern matching

The face center can be approximated by reusing information from the optical flow estimation, because the region around the eyes and the nose shows a characteristic flow pattern. A template containing the flow information of an average face center is created offline. The face center's position and the directions it moves into, are retrieved by matching this template in specific ways with the optical flow image $\mathrm{OF}_{\mathrm{im}}$ (see Eq. (11)) of any sequence. An example for such a template is visualized in Fig. 6. The first image displays $|v_x|$ of a horizontal only movement whereas the second one shows $|v_y|$ taken from a vertical only motion, at the face center. The complete template $\mathrm{OF}_{\mathrm{temp}}$ used for the first matching is complex, combining v_x and v_y (see also Eq. (12)). Its absolute value is displayed in the third image of Fig. 6. We denote:

$$\underset{x,y\in T}{\mathrm{OF}_{\mathrm{temp}}} = v_x(x,y) + \mathrm{i}\cdot v_y(x,y) \tag{12}$$

where T is the template, which has approximately 2–3% the size of $\mathrm{OF}_{\mathrm{im}}$. We start by calculating the absolute similarity of $\mathrm{OF}_{\mathrm{temp}}$ and $\mathrm{OF}_{\mathrm{im}}$, at pixels where $\mathrm{roi} = 1$, as in the following equation:

$$\frac{< |\mathrm{OF}_{\mathrm{im}}|, |\mathrm{OF}_{\mathrm{temp}}| >}{\|\mathrm{OF}_{\mathrm{im}}\| \cdot \|\mathrm{OF}_{\mathrm{temp}}\|} \leqslant 1 \tag{13}$$

resulting in a similarity matrix sim with values in $[0,1]$. The value max(sim) is stored in f_{cer} and two further similarities are calculated at that position:

$$-1 \leqslant \frac{< \Re(\mathrm{OF}_{\mathrm{im}}), \Re(\mathrm{OF}_{\mathrm{temp}}) >}{\|\Re(\mathrm{OF}_{\mathrm{im}})\| \cdot \|\Re(\mathrm{OF}_{\mathrm{temp}})\|} \leqslant 1 \tag{14a}$$

$$-1 \leqslant \frac{< \Im(\mathrm{OF}_{\mathrm{im}}), \Im(\mathrm{OF}_{\mathrm{temp}}) >}{\|\Im(\mathrm{OF}_{\mathrm{im}})\| \cdot \|\Im(\mathrm{OF}_{\mathrm{temp}})\|} \leqslant 1. \tag{14b}$$

Eqs. (14a) and (14b) give a scalar in $[-1,1]$ referred to as $\mathrm{sim}_{\mathrm{h}}$ and $\mathrm{sim}_{\mathrm{v}}$, respectively. These similarity measures indicate the directions (by their signs) in which the face center moves and the relative velocities, e.g. whether the actual movement is more a horizontal than a vertical one.

The procedure described in this section can effectively be employed to detect the center facial region, whereas it is less applicable to detect the boundary facial features. Furthermore it is a quickly computed attention center for subsequent analysis.

4.2. Model-based Gabor feature extraction

The second method presented for face part detection is similar to [13]. Essentially, features are extracted at certain points of a non-uniform "retinotopic" grid in order to measure image properties of a face [12,28–30]. These features constitute models, used for SVM classifiers [31] during training and in the operational phase, when the system automatically locates the face. In Fig. 7, two out of the three employed facial models are displayed. Each model uses specific points chosen out of the retinotopic grid, which are marked by plus signs. The center model

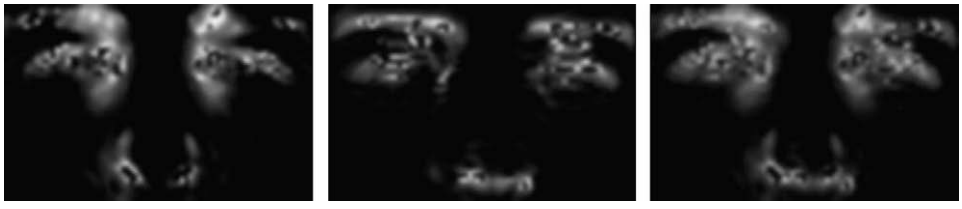

Fig. 6. Example face center template for OF pattern matching: horizontal OFL (left); vertical OFL (middle); magnitude of combined OFL (right) (Gabor-filtering used [5]).

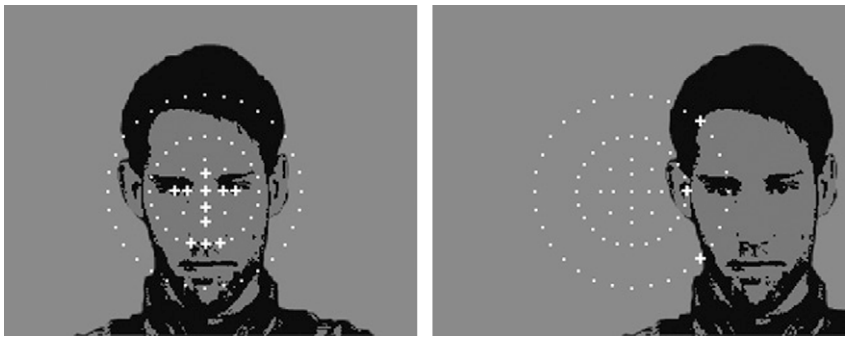

Fig. 7. Chosen points from a retinotopic grid to fit the center (left image) and side (right image) facial regions. Gabor features are applied at these points for modelling.

points (on the left-hand side) are denser, targeting also the middle of the face, whereas the outer model points (on the right-hand side) are chosen to be bent along the facial curvature. At these points, features are extracted by means of a Gabor filter bank, which in our case consists of five (radial or isotropic) frequency and six orientation channels. The filter bank is designed in the log-polar domain (a logarithmically scaled polar space), where the Gabor filters are uniformly distributed Gaussian bells [9]. This ensures that the designed Gabor filters evenly cover the Fourier domain. Only specific frequency and orientation channels are used (see Table 1) within the models, because the approximate distance range of the faces was assumed to be known. A further reason to pick out specific features at only some points of the grid is to speed up the feature extraction process. Like in the OFL approach, where we concentrate on horizontal and vertical lines, we select the Gabor filter orientations accordingly. The selected frequency channels are tuned to the scale of the features we are interested in. The outer models encode features taken at low frequency channels in order to cover a large area. The fine details of a face's center (eyes and nose) are modeled with high frequency channels. This adaption is done to focus on few but important features. The classification performance generally deteriorates when the number of features increases, if no dimension reduction is done e.g. due to the increased difficulties (i) to obtain sufficient data for statistical models (curse of dimensions), and (ii) to implement real-time processing [13,20].

The feature vector $\vec{k}$ for a point p of a specific model (a point marked with the + in Fig. 7) consists of single scalar products between the image and Gabor filters at the frequencies and orientations listed in Table 1. Accordingly, we obtain:

$$k_{(\xi,\eta)} = \left| \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} \mathrm{Im}_{(m,n)} f_{(m,n,\xi,\eta)} \right| \tag{15}$$

where a filter magnitude answer $k_{\xi,\eta}$ is formed by the absolute value of the scalar product of an image patch Im with size $M \times N$ and a complex Gabor filter f, of the same size. The size of this image patch around p depends on the frequency ξ, i.e. a higher frequency implies a smaller neighborhood and vice-versa. A single Gabor filter, denoted as $f_{m,n,\xi,\eta}$ is a 2D complex valued filter corresponding to a certain frequency ξ and orientation η. The resulting feature vector $\vec{k}$ consists of the magnitude answers $k_{\xi,\eta}$, which describe the neighborhood of the image at a point p. The dimension of $\vec{k}$ equals the product of the employed amount of frequencies and orientations (see Table 1). The complete feature vector $\vec{x}$ contains the elements of $\vec{k}$ of all grid points within a certain model. Finally $\vec{x}$ is classified by the corresponding SVM expert to deliver a certainty measure, either of g_{cer1}–g_{cer3} (in short $g_{\mathrm{cer1-3}}$ for the list of certainties of center, right and left face part expert).

To add face generalisation, the three experts (center, right, left) are trained with the last 145 frontal face images from the XM2VTS Database [32] and further 150 shots from a different camera [33]. All images are downsized to a resolution of 300×240 and 320×240, respectively, in order to reduce the computation time. The three experts are prepared for the training separately by manually marking positive and negative class examples (features) for the models' grid points in all training frames. Having these labels, non-linear SVM classifiers employing an RBF kernel are trained for each model. To achieve optimal classification rates, each classifier's kernel parameters are determined by twofold cross validation. The according classification rates can be found in Table 1.

If the optical flow pattern matching method described previously is preceding the Gabor feature based method, the critical search area for the latter can be reduced significantly. Additionally, like in the former method, only pixels where roi = 1 are considered as candidate positions for face parts, reducing the points at which to calculate $\vec{k}$ substan-

Table 1
Frequency and orientation channels used for face part models plus achieved cross-validation results over the training set

Model	Frequency channel	Orientation channel	Classification rate
Center	4,5	1,4	0.96
Right-outer	3,4	1,2,5	0.98
Left-outer	3,4	1,2,5	0.99

tially. Without the optical flow matching step, this method would not be restricted to detect only patterns in motion.

5. Liveness detection

In this section, the algorithm employed to detect liveness in a face image sequence of three frames is presented. It involves the concept described in the previous chapters. The flow chart in Fig. 8 shows all components and their interconnection. The enumeration in the following summary refers to the steps in Fig. 8:

(1) The OFL at the center image Im_2 is calculated using the algorithm described in Section 3.2. As an output we obtain the optical flow image, $\mathrm{OF}_{\mathrm{im}}$.
(2) The face center is detected by means of optical flow pattern matching as described in Section 4.1. The output of this step is the face center position (x_1/y_1) including a certainty measure f_{cer}. Additionally we obtain the directions $\mathrm{sim}_{\mathrm{h}}$ and $\mathrm{sim}_{\mathrm{v}}$ describing the central motion.
(3) All three face parts are detected in the center image Im_2 by model-based Gabor feature classification, outlined in Section 4.2. As the position of the face center is already known from step 2, the feature extraction is only employed in a small neighborhood of the size 10×10. The two outer models are extracted each in the expected neighborhood to the right and left of the face center. As a result we obtain the face part coordinates (x_2/y_2), (x_3/y_3) and the certainties $g_{\mathrm{cer1-3}}$. The purpose is to confirm the face center position detected rapidly in step 2 and to ensure the presence of an actual face. If the center model expert's answer is below a certain threshold, the position x_1/y_1 is discarded and a saccadic image search for the face center is done by local Gabor decomposition. This involves that the output of step 2 (f_{cer}, $\mathrm{sim}_{\mathrm{h}}$ and $\mathrm{sim}_{\mathrm{v}}$) have to be recalculated at the new position as well. Though possible, such a case occurs very rarely. It is worth noting that the instantly derived roi is input to steps 1–3 and is used to speed up the procedure.
(4) In this step, the certainties f_{cer} and $g_{\mathrm{cer1-3}}$ are verified and updated if necessary. If one of them is below a common threshold τ, the sequence is regarded to be unsuitable due to an insufficiently recognizable face (non-face), yielding a liveness score of 0.
(5) A rectangular area around each face part's (central) position x_{1-3}/y_{1-3} is cut out of $\mathrm{OF}_{\mathrm{im}}$ and stored as image parts: $\mathrm{OF}_{\mathrm{part1}}$ (here 40×40) is situated around the face center, whereas $\mathrm{OF}_{\mathrm{part2}}$ (here 20×40) and $\mathrm{OF}_{\mathrm{part3}}$ (here 20×40) are located around the right- and left-hand side face part position, respectively. In Fig. 2, these three regions are indicated.
(6) Finally $\mathrm{OF}_{\mathrm{part1-3}}$ are compared to each other in order to deliver the liveness score. Only the values of each $\mathrm{OF}_{\mathrm{part}}$ greater than half its maximum absolute value are considered. This is to concentrate on high velocities only and to eliminate the negative impacts of still regions. The remaining values are divided by their total number in order to prepare for summation in their mean value calculation. We decide to concentrate on the primary movement only. In step 2, we retrieve the direction of the central motion $\mathrm{sim}_{\mathrm{h}}$ and $\mathrm{sim}_{\mathrm{v}}$. A mainly horizontal motion is indicated by $|\mathrm{sim}_{\mathrm{h}}|$ being larger than $|\mathrm{sim}_{\mathrm{v}}|$, otherwise suggesting vertical motion. The ratios cr and cl, which contribute to the final liveness score are calculated as follows:if $|\mathrm{sim}_{\mathrm{h}}| > |\mathrm{sim}_{\mathrm{v}}|$

$$\mathrm{cr} = \frac{\sum \Re(\mathrm{OF}_{\mathrm{part1}})}{\sum \Re(\mathrm{OF}_{\mathrm{part2}})}, \qquad \mathrm{cl} = \frac{\sum \Re(\mathrm{OF}_{\mathrm{part1}})}{\sum \Re(\mathrm{OF}_{\mathrm{part3}})}$$

else

$$\mathrm{cr} = \frac{\sum \Im(\mathrm{OF}_{\mathrm{part1}})}{\sum \Im(\mathrm{OF}_{\mathrm{part2}})}, \qquad \mathrm{cl} = \frac{\sum \Im(\mathrm{OF}_{\mathrm{part1}})}{\sum \Im(\mathrm{OF}_{\mathrm{part3}})}.$$

Depending on the primary movement, the ratios cr and cl compare the real or the imaginary part of $\mathrm{OF}_{\mathrm{part1}}$ with $\mathrm{OF}_{\mathrm{part2}}$ and $\mathrm{OF}_{\mathrm{part3}}$, respectively. A ratio between the center and a side motion having an absolute value greater than 1, indicates liveness. In the ideal case, we expect the face center part moving in an opposite direction compared to the sides, which is indicated by negative ratios. By including this observation, the liveness score is divided into a

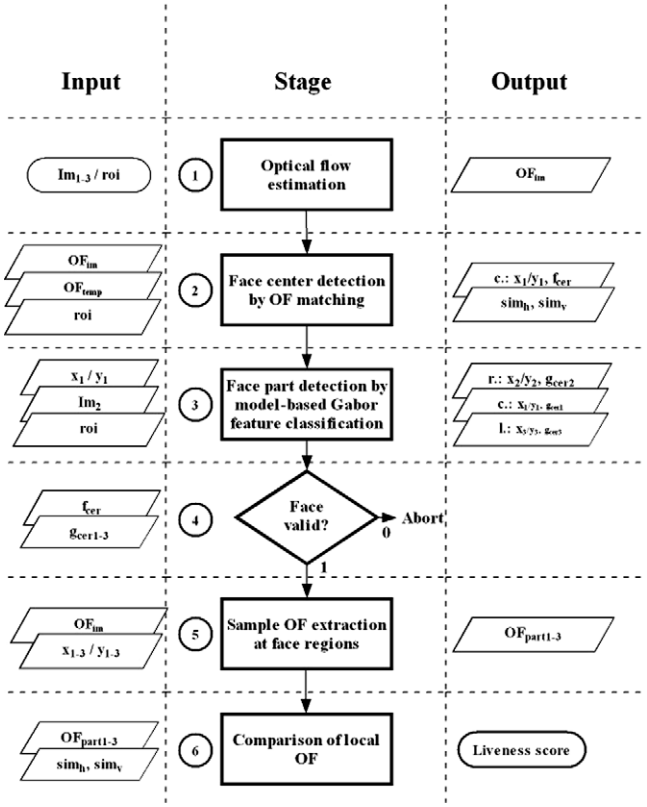

Fig. 8. Flow chart detailing the liveness detection.

velocity dependent and a direction dependent part. The final liveness score is then constructed as,

$$L = \frac{1}{4}[(|\mathrm{cr}| > \tau) + (|\mathrm{cl}| > \tau) + (\ominus \mathrm{cr} < 0) + (\ominus \mathrm{cl} < 0)]$$

where $\ominus x$ means "sign of" x and takes either of the values -1 or 1. Note, that τ is equal to 1 according to the above reasoning. However, we are allowed to set τ to higher values, e.g. 1.5–2, to further decrease the probability of false accepts if this does not affect the false rejection substantially. The score is a value in $[0,1]$, where 0 indicates no liveness and 1 represents the maximum liveness.

6. Experiments

To evaluate the performance of our liveness detection scheme, we use the "Head Rotation Shot"-subset (DVD002 media) of the XM2VTS database. Furthermore, we perform "real" experimental spoofing by manually playing through plausible attack scenarios (so-called playback attacks) involving a (different) camera at our laboratory [33]. In case of the database tests we employ the first 100 videos from the first session only. These are downsized from 720×576 to 300×240 pixels. For experiments with the laboratory camera a resolution of 320×240 pixels is used.

6.1. Live and non-live sequences

For the "live sequences", which are expected to obtain a high liveness score, four sequences (of either three or five frames)[1] containing partial rotation are cut from each video in case of the database experiments. The possible movements (to the left, to the right, up, down) are evenly present in the sequences. On the other hand, "non-live sequences" have to be manufactured because a database for playback attacks using photographs is not available: for each person from the database one frame is taken out of a respective live sequence, and translated each horizontally and vertically to produce sequences of three or five frames. This yields four non-live sequences per person, imitating high resolution photographs in motion. Two live sequences on top of their non-live counterparts are displayed in Fig. 9. The first two rows contain a horizontal movement, whereas it is a vertical one in the last two rows. Note that the motion in the bottom live sequence is hardly noticeable by the human eye. Furthermore, four different playback attack scenarios are constructed (see Section 6.3). Here, "non-live sequences" consist of three images (consecutive frames) captured by the laboratory camera. For the first three spoofing trials we use high quality printouts which are moved in different manner in front of the camera. In order to show the limits of the proposed liveness detection we include a portable video device in the final spoofing trial. For the discussed reason, τ was set to 1.5 when calculating the liveness scores.

[1] We also wanted to study the effects of using more than three frames.

6.2. Database test results

A total of 400 live- and 400 non-live sequences were analyzed by the liveness detection system described in the previous chapter. The liveness score achieved by each sequence was stored as primary result. In addition to that, the progression of the face part detection was monitored for each sequence. The system performance is visualized as DET (detection error tradeoff) curves in Fig. 10, in case of using three and five frames for the OFL, on the left- and right-hand side, respectively. Also, the impact of τ is shown for three discrete values. What we can immediately observe, looking at Fig. 10 is, that the EER (equal error rate) is 0.5% in case of using three frames, whereas it is $\geqslant$0.5% when using five frames. Inspecting the scores more closely reveals that most of the non-live sequences scored 0, whereas most of the live sequences achieved a liveness score of 0.75. Only 1 non-live sequence got a score of 0.5, to be discussed further below. This single sequence is also responsible for the FA (false acceptance) plateau at 0.5%. As expected, scores are pushed downwards if τ increases, resulting in higher FR (false rejection) but also lowering FA (false acceptance) rates. In the case of using five frames, scores are distributed from 0.75 towards 0.5 and 1, and even 0.25. It is to be concluded that while the motion estimation is in favor of the five frames, particularly for non-live sequences, non-uniform head motion is not, and the use of three frames is recommended. Fig. 11 shows the single non-live sequence for which the method failed, i.e. the sequence was given a liveness score of 0.5. Note, that we actually display the first (leftmost) and last frame (rightmost) of the sequence while the center frame (middle) is replaced by the OFL in a direction, here vertical as the sequence contains a top–down translation. Additionally the focused face parts (center, right, left) which are automatically detected in the center frame are indicated by rectangles in the OFL image. As can be observed, no horizontal line structure is available in the observed side areas of the face, leading to non-measurable (vertical) velocities. On the contrary, Fig. 12 shows a typical live sequence ($L = 1$) with some horizontal motion (of vertical lines) in it. Inspecting the outcomes, we also found that eyeglasses can lower the liveness score of live sequences, as they are near the camera even at the sides.

The success rate of the alternative face center detection (by OF patterns) was further encouraging, as it was 92% in isolation. The main reasons for false localization were eyeglasses and moustaches, which could generate distracting flow patterns. Wrongly located face centers were however autonomously rejected by the model-based method, which could successfully assist to locate the face center in the remaining 8% of the sequences. All side-regions were located sufficiently correctly.

Fig. 9. Rows 1 and 3 show example live sequences, rows 2 and 4 display the according non-live sequences (playback attack).

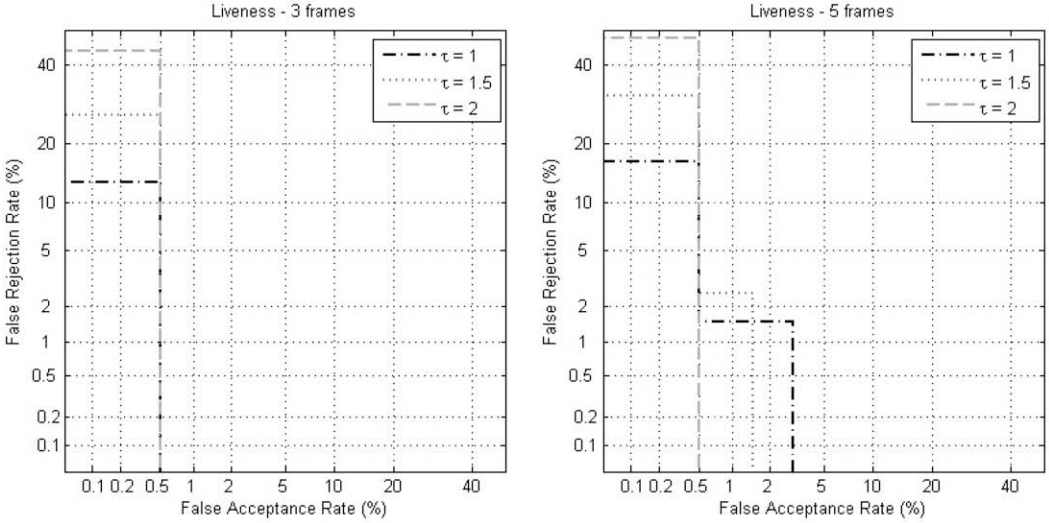

Fig. 10. DET curves for the liveness detection, using (i) three frames and (ii) five frames for the optical flow estimation. The EER is to be read off at the intersection point of a curve with the diagonal line.

6.3. Applied spoofing results

In our first spoofing trial we moved a high quality printout horizontally in front of the laboratory camera, which basically amounts to the foregoing experiments, verifying their practical value. The analogous images are displayed Fig. 13. Looking at the OFL image in Fig. 13 we can observe that the face undergoes constant motion, which is against

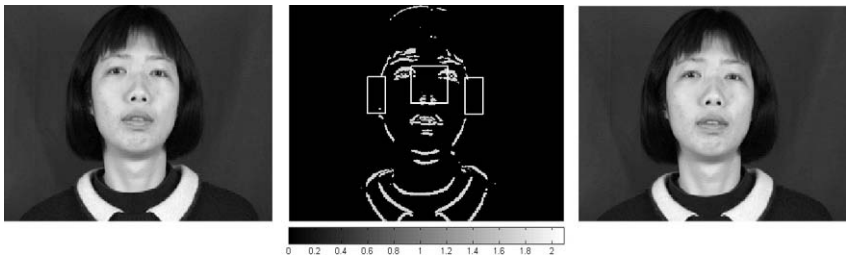

Fig. 11. A non-live sequence that posed difficulties for the system: frame 1/3 (left); vertical OFL (middle); frame 3/3 (right).

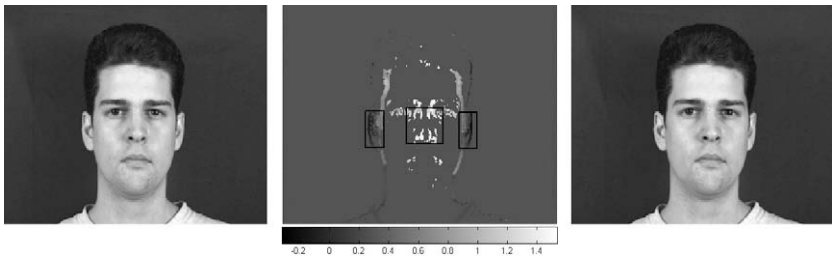

Fig. 12. Typical live sequence: frame 1/3 (left); horizontal OFL (middle); frame 3/3 (right).

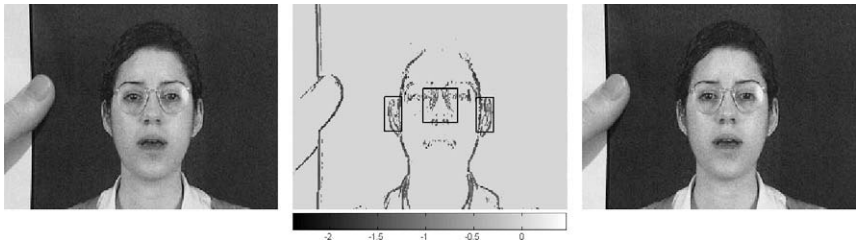

Fig. 13. Translated photograph: frame 1/3 (left); horizontal OFL (middle); frame 3/3 (right).

our definition of liveness and leads to a liveness score of 0. One might argue that it would be natural to fool the system by bending the surface of the printout with the aim of imitating a 3D face. To address this point, we proceeded in two ways. In Fig. 14, the first spoofing attempt is visualized. As can be observed, the focused motion is still fairly constant although the image surface is bent. In case of a translated photo, it is actually natural that the face boundaries represent significant edges, which are likely to generate strong motion. In order to better mimic a live face, we wrapped a photograph around a cone and rotated it in horizontal direction. For the particular sequence displayed in Fig. 15, even less motion was measured in the face center compared to the outer face parts ($L = 0$), which should not be taken as a general statement though. A photo wrapped around a cylinder (Fig. 15) is different from a real head, since it is not as peaked, and the sides (ears) will not move into opposite directions compared to the face center (thus not yielding liveness scores above 0.5). The motion may be higher in the center, which we can react upon by setting τ to a higher value (e.g. 2), and also, the face detection will fail if a face (photo) looks too distorted.

However, for the purpose of showing system limitations we recorded a video of a person with the laboratory camera. Then the video was replayed on a portable video player equipped with a high quality display positioned in

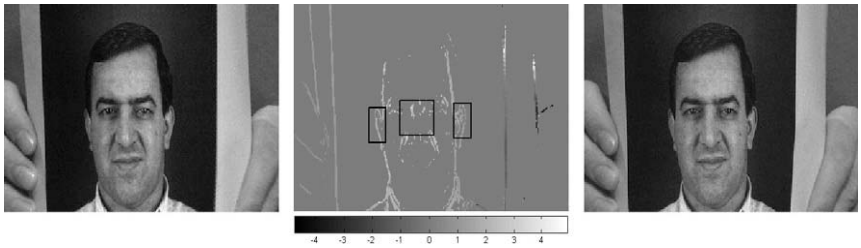

Fig. 14. Bent photograph in motion: frame 1/3 (left); horizontal OFL (middle); frame 3/3 (right).

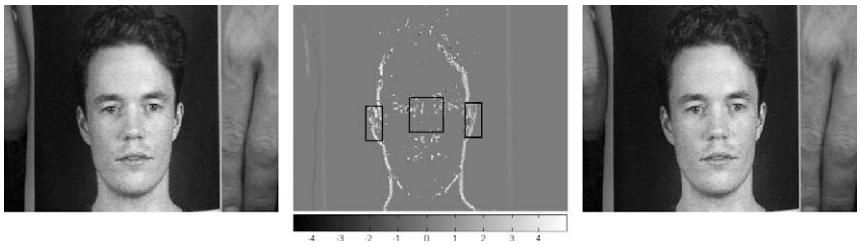

Fig. 15. Photograph wrapped around a cone and rotated: frame 1/3 (left); horizontal OFL (middle); frame 3/3 (right).

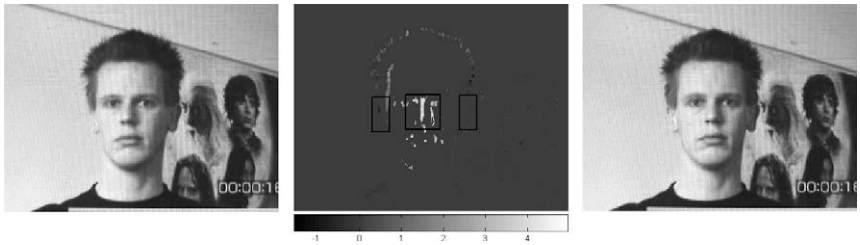

Fig. 16. Video playback: frame 1/3 (left); horizontal OFL (middle); frame 3/3 (right).

ideal distance. The outcome of such a playback attack is shown in Fig. 16. Here, a liveness score of 0.75 was calculated, which is no surprise since the video contains exactly the liveness indicators our method relies on. An analysis of the image quality of such a sequence could expose further useful features for liveness detection. Also, any rectangles or constant moving background seem suspicious and should be considered in future work. Another possible spoofing method we can think of able to fool the proposed scheme is wearing a mask. However, it is then mainly task of the face part detection (face recognition) stage to reject such a claim, since also a human expert could be deceived. We do not give further quantitative results in this second experimental part, since many issues are highly speculative and subjective. A quantification of the success for these new spoofing techniques is difficult to measure as they assume a successful chain of extremely difficult and rare outcomes to cooperate. For example, the impostor must find out the focal length of the camera (which is not public information) to place the video display at the perfect distance and without first showing anything else than the video display content.

7. Discussion

The explanation for non-live sequences achieving a score greater than 0 lies in up/down-movement, in case there is less horizontal line structure to observe at the side parts of a face, compared to the face center. If there are no or few lines present inside a checked side-region, the velocity at the face center may be measured to be higher. It is worth noting that the system would have been error-free on the test set if only sequences containing horizontal movement had been considered, with a big margin. Although our results support the general case, we suggest to focus on horizontal movements only in liveness detection systems of the proposed kind, e.g. demanding $|\mathrm{sim_h}| \gg |\mathrm{sim_v}|$ before evaluating the score.

Furthermore, it is worth mentioning that the last scenario was especially difficult to construct since the camera had to zoom in and focus the screen of the portable video player perfectly, meaning that the experimental conditions had to be changed, otherwise not making the spoofing possible in the first place. Disabling the zoom would force anyone to bring along a big (1:1) video display to achieve such a liveness score. Also, the device had to be kept absolutely still to not affect the recorded motion.

Background or illumination changes can affect the success of the proposed scheme in that the face part detection could fail. Given a less controlled environment, the face part experts would have to be retrained. Despite the results, a more continuous extraction of the OFL could have been advantageous, too. For this reason we plan to investigate alternative face detection techniques in the future. An advantage of the proposed liveness detection scheme is its non-intrusiveness, i.e. it does not require any user interaction. To avert video spoofs we recommend to demand and examine user reactions, e.g. analyzing the lip movement in a text prompted scenario where a user must utter something which is randomly presented to him.

8. Conclusion

Evaluating the trajectory of several face parts using the optical flow of lines is the main novelty of the proposed system. The liveness detection is successful in separating live face sequences from still photographs in motion, with an equal error rate of 0.5% on the test data. Further investigation into practical application confirmed the robustness of the scheme. Although restricted to line velocity estima-

tion, the suggested OFL is able to deliver robust measurements for face liveness assessment. The Gauss-based implementation is both efficient and effective. With regard to face localization, a quick method utilizing optical flow patterns was shown feasible as well. Additionally, the face part detection by model-based Gabor feature classification is robust to typical sources of errors like glasses and facial hair, and it effectively monitors and complements the previous method. We were also able to reduce the model complexity by adaptation of the features in frequency and orientation, which translated into a speed-up. The scheme was evaluated on the XM2VTS database having scale variations up to 10%.

References

[1] N.K. Ratha, J.H. Connell, R.M. Bolle, Enhancing security and privacy in biometrics-based authentication systems, IBM Syst. J. 40 (2) (2001) 614–634.

[2] S.A.C. Schuckers, Spoofing and Anti-Spoofing Measures, Information Security Technical Report 7 (4) (2002) 56–62.

[3] T. Choudhury, B. Clarkson, T. Jebara, A. Pentland, Multimodal person recognition using unconstrained audio and video, in: 2nd International Conference on Audio-Visual Biometric Person Authentication, Washington DC, 1999.

[4] J. Li, Y. Wang, T. Tan, A.K. Jain, Live face detection based on the analysis of Fourier spectra, in: Biometric Technology for Human Identification, SPIE, vol. 5404, 2004, pp. 296–303.

[5] K. Kollreider, H. Fronthaler, J. Bigun, Evaluating liveness by face images and the structure tensor, in: Fourth IEEE Workshop on Automatic Identification Advanced Technologies AutoID 2005, Buffalo, New York, 2005, pp. 75–80.

[6] J. Bigun, G.H. Granlund, J. Wiklund, Multidimensional orientation estimation with applications to texture analysis and optical flow, IEEE-PAMI 13 (8) (1991) 775–790.

[7] D. Gabor, Theory of communication, J. IEE 93 (1946) 429–457.

[8] H. Knutsson, Filtering and reconstruction in image processing, PhD Thesis No: 88, Linköping University, ISY Bildbehandling S-581 83 Linkoeping, 1982.

[9] J. Bigun, Speed, frequency, and orientation tuned 3-D Gabor filter banks and their design, in: Proc. International Conference on Pattern Recognition, ICPR, Jerusalem, IEEE Computer Society, 1994, pp. C–184–187.

[10] J. Barron, D. Fleet, S. Beauchemin, Performance of optical flow techniques, Inf. J. Comput. Vis. 12 (1) (1994) 43–77.

[11] F. Smeraldi, J. Bigun, Facial features detection by saccadic exploration of the Gabor decomposition, in: International Conference on Image Processing, ICIP-98, Chicago, October 4–7, vol. 3, 1998, pp. 163–167.

[12] F. Smeraldi, J. Bigun, Retinal vision applied to facial features detection and face authentication, Pattern Recogn. Lett. 23 (2002) 463–475.

[13] J. Bigun, H. Fronthaler, K. Kollreider, Assuring liveness in biometric identity authentication by real-time face tracking, in: CIHSPS2004 – IEEE International Conference on Computational Intelligence for Homeland Security and Personal Safety, Venice, Italy, IEEE Catalog No. 04EX815, ISBN 0-7803-8381-8, 2004, pp. 104–112.

[14] M. Lades, J.C. Vorbruggen, J. Buhmann, J. Lange, C.v.d. Malsburg, R.P. Hurtz, W. Konen, Distortion invariant object recognition in the dynamic link architectures, IEEE Trans. Comput. 42 (3) (1993) 300–311.

[15] B. Duc, S. Fischer, J. Bigun, Face authentication with Gabor information on deformable graphs, IEEE Trans. Image Process. 8 (4) (1999) 504–516.

[16] G.A. Orban, Neuronal Operations in the Visual Cortex, Studies of Brain Functions, Springer, 1984.

[17] D.H. Hubel, T.N. Wiesel, Receptive fields of single neurones in the cat's striate cortex, J. Physiol. (London) 148 (1959) 574–591.

[18] S. Marcelja, Mathematical description of the responses of simple cortical cells, J. Opt. Soc. Am. 70 (1980) 1297–1300.

[19] A.L. Yarbus, Eye Movements, Plenum, New York, 1967.

[20] I.R. Fasel, M.S. Bartlett, J.R. Movellan, A comparison of Gabor methods for automatic detection of facial landmarks, in: International Conference on Automatic Face and Gesture Recognition, 2002, pp. 242–248.

[21] E.S. Bigun, J. Bigun, B. Duc, S. Fischer, Expert conciliation for multi modal person authentication systems by Bayesian statistics, in: J. Bigun, G. Chollet, G. Borgefors (Eds.), Audio and Video based Person Authentication – AVBPA97, Springer, 1997, pp. 291–300.

[22] A. Jain, L. Hong, Y. Kulkarni, A multimodal biometric system using fingerprint, face and speech, in: Audio and Video based Person Authentication – AVBPA99, 1999, pp. 182–187.

[23] R.W. Frischholz, U. Dieckmann, BioID: a multimodal biometric identification system, IEEE Comput. 33 (2) (2000) 64–68.

[24] G. Chetty, Multi-level liveness verification for face-voice, in: Biometrics Symposium, 2006.

[25] G.H. Granlund, In search of a general picture processing operator, Computer Graph. Image Process. 8 (2) (1978) 155–173.

[26] J. Bigun, G.H. Granlund, Optimal orientation detection of linear symmetry, in: First International Conference on Computer Vision, ICCV, June 8–11, London, IEEE Computer Society Press, Washington, DC, 1987, pp. 433–438.

[27] J. Bigun, Vision with Direction, Springer, 2006.

[28] F. Smeraldi, O. Carmona, J. Bigun, Real-time head tracking by saccadic exploration and Gabor decomposition, in: A.T. Almeida, H. Araujo (Eds.), Proc. the 5th International Workshop on Advanced Motion Control, vol. IEEE Cat. No. 98TH8354, IEEE Service Center, 445 Hoes Lane, P.O. Box 1331 Piscataway, NJ 08855-1331, USA, 1998, pp. 684–687.

[29] M. Tistarelli, G. Sandini, On the advantages of log-polar mapping for direct estimation of time-to-impact from optical flow, IEEE Trans. Pattern Anal. Mach. Recogn. 15 (4) (1993) 401–410.

[30] J. Bigun, Gabor phase in boundary tracking and region segregation, in: Proc. DSP & CAES Conf. Nicosia, Cyprus, University of Nicosia, 1993, pp. 229–237.

[31] C. Cortes, V. Vapnik, Support-vector networks, Mach. Learn. 20 (1995) 273–297.

[32] K. Messer, J. Matas, J. Kittler, J. Luettin, G. Maitre, XM2VTSDB: The extended M2VTS database, in: Audio and Video based Person Authentication – AVBPA99, University of Maryland, 1999, pp. 72–77.

[33] Specification Sheet of Sony Pan/Tilt/Zoom Network Camera SNC-RZ30, available from: http://www.visualsecurity.com/html/sonysncrz30.html.