Real-Time Face Detection and Motion Analysis With Application in "Liveness" Assessment

Klaus Kollreider, Hartwig Fronthaler, Maycel Isaac Faraj, and Josef Bigun

Kollreider K, Fronthaler H, Faraj M, Bigun J. Real-Time Face Detection and Motion Analysis With Application in "Liveness" Assessment. IEEE; IEEE Transactions on Information Forensics and Security. 2007;2(3 part 2):548-558.

DOI: http://dx.doi.org/10.1109/TIFS.2007.902037
Copyright: IEEE

Real-Time Face Detection and Motion Analysis With Application in “Liveness” Assessment

Klaus Kollreider, Hartwig Fronthaler, Maycel Isaac Faraj, and Josef Bigun, *Fellow, IEEE*

***Abstract*—A robust face detection technique along with mouth localization, processing every frame in real time (video rate), is presented. Moreover, it is exploited for motion analysis onsite to verify “liveness” as well as to achieve lip reading of digits. A methodological novelty is the suggested quantized angle features (“quangles”) being designed for illumination invariance without the need for preprocessing (e.g., histogram equalization). This is achieved by using both the gradient direction and the double angle direction (the structure tensor angle), and by ignoring the magnitude of the gradient. Boosting techniques are applied in a quantized feature space. A major benefit is reduced processing time (i.e., that the training of effective cascaded classifiers is feasible in very short time, less than 1 h for data sets of order 10^4). Scale invariance is implemented through the use of an image scale pyramid. We propose “liveness” verification barriers as applications for which a significant amount of computation is avoided when estimating motion. Novel strategies to avert advanced spoofing attempts (e.g., replayed videos which include person utterances) are demonstrated. We present favorable results on face detection for the YALE face test set and competitive results for the CMU–MIT frontal face test set as well as on “liveness” verification barriers.**

***Index Terms*—AdaBoost, antispoofing, face detection, landmark detection, lip reading, liveness, object detection, optical flow of lines, quantized angles, real-time processing, support vector machine (SVM).**

I. Introduction

In face analysis related to biometrics, one may distinguish between face detection and face recognition. The former deals with locating one or multiple instances of human faces in a photograph or video whereas the latter targets on establishing a unique link between two or more (independent) face recordings of the same person. It is worth noting that face detection is a prerequisite to both subcategories of face recognition: Authentication (1:1 matching) and identification (1:n matching). This is because an accurate registration between a pair of face images, and/or between the landmarks (e.g., eyes and mouth) of the face images, has a pivoting role on the way toward good recognition performance. A realistic application hardens the task. These considerations also imply that any image region deemed as face—even nonfaces—will be considered, ignoring the fact that the identity establishment can be compromised prior to authentication/identification, because nonfaces are given a real chance to impost. Additionally, material for counterfeiting any biometrics is easy to acquire (e.g., photographs from a website, or traces of fingerprints left on a glass) and, thus, raising antispoofing barriers is indispensable at the very start of biometric person recognition. Methods bringing benefits to both face and “liveness” detection are desirable, which is also the application focus of this paper.

A. Face/Landmark Detection

When attempting to detect faces (or locate a single face) in a visual representation, image-based and landmark-based methods may be primarily distinguished between [1] and [2]. This paper focuses on the detection of frontal faces in 2-D images although we will be using a sequence of them to assess 3-D information for “liveness” assessment purposes. The features here represent measurements made by means of some basis functions in a multidimensional space which should be contrasted to the term “facial features” sometimes used in the published studies to name subparts of a face (e.g., the eyes, mouth, etc.). We will call the latter “facial landmarks” or “landmarks.” Challenges in face detection generally comprise varying illumination, expression changes, (partial) occlusion, pose extremities [1], and requirements on real-time computations.

The main characteristics of still image-based methods are that they process faces in a holistic manner. That is, no parts of the face are intentionally favored to be used for face detection, or when a favoring is undertaken, the selection of face parts is left to the training/classifier. Faces are learned by training on roughly aligned portraits as well as nonface-like images. The training is typically part of a high-level statistical pattern recognition method [e.g., a Bayesian classifier, an artificial neural network (ANN), or a support vector machine (SVM)]. When operational, the trained classifier provides a decision on being face or nonface at subparts of an image and at various scales. A popular approach, although primarily designed for face recognition, uses the so-called Eigenfaces [3], or the principal component analysis (PCA) coordinates to quantify the “faceness” of an image (region). More recent face detection systems include [4] and [5] which use neural networks, whereas SVMs [6], were employed in [7] to classify image regions as a face or nonface. A naive Bayes scheme was implemented in [8], and recently in [9], whereas an AdaBoost procedure [10] was concurrently adapted in [11] and [12] for the purpose of classification of regions with respect to “faceness.” Image-based methods can easily be trained in an analogous manner for the purpose of detecting other objects in images (e.g., cars). The AdaBoost-based face detection in [11] and [12] has been suggested as being real

Manuscript received November 1, 2006; revised April 14, 2007. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Vijaya Kumar Bhagavatula.

The authors are with Halmstad University, Halmstad SE-30118, Sweden (e-mail: klaus.kollreider@ide.hh.se; hartwig.fronthaler@ide.hh.se; maycel.faraj@ide.hh.se; josef.bigun@ide.hh.se).

Digital Object Identifier 10.1109/TIFS.2007.902037

time, and has been followed up by other studies extending it to multiposes, and reducing classifier complexity (e.g., [13] and [14]). However, the employed features play a decisive role besides the used classifiers. Of all published methods, only a few use the gray values directly as features to be classified. However, almost all approaches use a preprocessing of the gray values (e.g., histogram equalization or normalization) to minimize the effect of adverse light conditions at the expense of computational processing. The methods suggested by [11], [13], and [14] use Haar-like rectangle features, translating into high detection speed whereas [9] and [12] employed edge features with arguably lower execution speed. As will be detailed further, some of the advantages of those methods can also be identified in the scheme we suggest here.

A novelty in this study is the use of gradient angles only, allowing the gray value preprocessing to be expendable. Also, the use of hierarchical and adaptive quantization levels improves the detection performance as opposed to a few and fixed angle quantizations (e.g., seven in [9]). Another contribution is the use of not only the gradient direction information, but in addition, the structure tensor direction [15] is exploited to encode the local structure. Since we use quantized angle features, we call the latter "quangles" for expediency. Furthermore, these quangles are boosted in layers of a decision cascade as in [11], also enabling small classifiers. Separable filtering and the use of lookup tables amount to the remaining speedup that results in efficient computations when the system is operational, as well as offline when it is used for training. We achieve scale invariance through signal theoretically correct downsampling in a pyramidal scheme. The usefulness of the method is shown in the context of face and landmark (mouth) detection. A methodological advantage of the suggested scheme is some readily availability filtered signals that are also highly desirable for other tasks, for example, real-time optical-flow calculations when implementing "liveness" [16] verification barriers in biometric person authentication. In comparison, the rectangle features suggested in [11], despite their value in pure object detection in still images, have limited reusability when it comes to motion tasks (e.g., motion quantification).

To have a fuller picture, we also summarize the basic rationale of landmark-based methods. Often citing biological motivations, they focus on a few salient face parts, landmarks at which most of the processing is concentrated. Such landmarks are, for example, the single eyes, mouth, nose (nostrils), eyebrows, etc. Many ideas have been published on how to extract those (e.g., using Gabor features and SVM, which may include techniques referred to in the former category, but employed in a local window. During training, face parts are primarily learned as opposed to the whole face (holistic). All candidate sites are examined according to local models but in a global scope by means of a (global) shape model. However, if more than one face is present in an image, the complexity quickly increases and landmark-based methods encounter computational problems when real-time performance is a demand. With currently available technologies, their use is hampered in such applications. Nevertheless, they are considered potentially more robust to partial occlusion and pose changes of a face as well as being more precise in localization. A survey of face detection methods using landmarks is given in [1] and [17].

B. "Liveness" Verification

In the literature, there are few "liveness" verification studies in connection with biometric person recognition, especially for face biometrics. Also, the commercial face recognition systems suffer from having a comprehensive strategy toward the problem of "liveness." Intuitively, one could use a multimodal system (e.g., face tracking and fingerprint [18]), with numerous sensors (e.g., several cameras, heat-sensitive cameras, etc.) to ease the task. Such configurations might not be applicable for a variety of reasons (e.g., due to being perceived intrusive or costly), and besides, being complex. One can instead avail the additional "liveness" information hidden in an image sequence (video). As an effort in this direction, methods exploiting changes in face expression [19] and landmark trajectories [16], [20] have been suggested. Although these methods are weak against video replay attacks, they can nevertheless be used as barriers against less intricate attacks (e.g., those using photographs). These days, an attacker could benefit from the advancing portable digital video player device technologies. Such a device positioned well enough in front of an unsophisticated camera system could pose a potential threat to break into a biometric recognition system. However, it is likely that an advanced "liveness" verification barrier would contain a reliable face detector and tracker as well as local motion (optical flow) estimators. Presented photographs can be rejected in a way that is shown in [16] by using onsite motion information. The main idea in this barrier is to accumulate evidence for the three-dimensionality of the (somehow) detected face(s) by employing the optical flow of lines (OFL) to conclude on the 3-D structure. Here, in contrast to [19], face expression changes are helpful, yet are not required because any head movements will suffice. In this study, we propose another barrier that uses lip movement classification and lip-reading for the purpose of "liveness" detection in a text-prompted dialog scenario. This means that the system will expect the person to utter something (e.g., a prompted random sequence of digits) and it should verify whether the observed lip dynamics fit in. For so-called "talking face" systems, this has also been responded to, most recently in [21], by inspecting the correlation of audio and video channels. The novelty in our study is that we suggest a method for real-time assessment of lip motion to recognize digits without audio information. A simple analysis of the mouth regions was performed in [21], whereas [9] focused on four corner points of the mouth/lips. On the other hand, tracking the lip contours as, for example, used in audio-visual recognition [22] is susceptible to noise, often requiring human intervention. We propose to automatically locate the mouth region and extract the enclosed OFL in real time. This implies that we model the motion of lips by moving line patterns in space-time planes [23]. We assume a digit-prompted scenario using an SVM expert to classify the lip dynamics. Lipreading by motion analysis has also been shown to be useful for person authentication [23], [24]. The scheme allows for a person to just whisper or to mime the digits, thereby counteracting eavesdropping.

We present experimental results on several public databases: The pure face detection is assessed on the MIT–CMU [5] and the YALE [25] face test sets, both being representative for real-world (and extreme light) conditions, also used by other studies to this end. We also provide experimental considerations concerning computational complexity, contrasting our method to Viola and Jones' approach [11] by using the open computer vision library (OpenCV). The "liveness" experiments are conducted on the XM2VTS database [26], simulating a digit prompted scenario in which the combined face and mouth detection is employed for autonomous processing.

II. Object/Face Detection

A. Quantized Angle Features (Quangles)

In this section, we present the features for object/face detection, which we call "quangles," representing quantized angle features. We exploit parts of the gradient information in order to determine the presence of a certain object. The gradient of an image is given in (1)

$$\nabla f = \begin{pmatrix} f_x \\ f_y \end{pmatrix} \tag{1}$$

where f_x and f_y denote the derivatives in the x and y direction, respectively. Furthermore, $|\nabla f|$ indicates the magnitude of the gradient and $\angle \nabla f$ refers to its angle. For the sake of object detection, we disregard the magnitude or intensity since it is highly affected by undesired external influences such as illumination variations (noise).

The key instrument of our quangle features is the quangle masks, which are denoted as follows:

$$Q(\tau_1, \tau_2, \phi) = \begin{cases} 1, & \text{if } \tau_1 < \phi < \tau_2 \\ 0, & \text{otherwise.} \end{cases} \tag{2}$$

The thresholds τ_1 and τ_2 constitute the boundaries of a partition in $[0, 2\pi]$. The quangle mask yields 1 if an angle ϕ is located within such a partition and 0 otherwise. In order to produce a set of quangle masks, we divide the full angle range $[0, 2\pi]$ into an increasing number of quantizations (partitions), which are additionally rotated. An example is depicted in Fig. 1. A set of quangle masks $\{Q\}_{\mathrm{maxQuant},\mathrm{nRot}}$ is fully determined by the maximum number of quantizations maxQuant and the number of rotations nRot. The parameter maxQuant has to be interpreted cumulatively, meaning that all quangle masks with less quantization steps are included in the set as well. The second parameter nRot indicates the number of rotations applied to each basic quangle mask. The final row in Fig. 1, for example, corresponds to $\{Q\}_{4,2}$, which consists of 27 different quangle masks. In order to create such a set of quangle masks, the thresholds τ_1 and τ_2 of each partition need to be determined. This can be done in a three-step procedure.

1) First, we define a sequence of threshold pairs α_1 and α_2 delimiting the desired number of partitions nQuant in the interval $[0, 2\pi]$, disregarding the rotational component

$$\alpha_1 = \frac{2\pi}{\mathrm{nQuant}} \cdot \mathrm{quant} \quad \alpha_2 = \frac{2\pi}{\mathrm{nQuant}} \cdot (\mathrm{quant} + 1) \tag{3}$$

where $\mathrm{quant} \in \{0, \ldots, \mathrm{nQuant} - 1\}$.

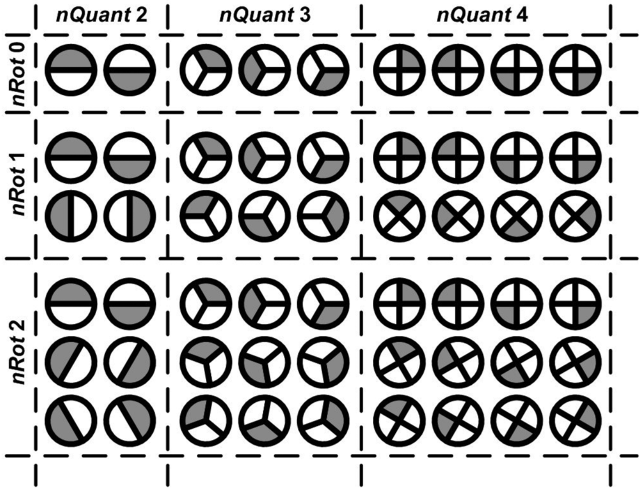

Fig. 1. Example of a set of quangle masks (angle displayed in polar form) using up to four quantizations and two rotations. The quangle masks defining larger partitions are automatically included in the set. The gray shaded areas correspond to the partitions yielding value 1 in (2).

2) In the second step, we create the final threshold sequence containing pairs of τ_1 and τ_2. For each partition quant, we include nRot rotated versions

$$\tau_k = \mathrm{mod}\left(\alpha_k - \frac{\pi}{\mathrm{nQuant}} \cdot \mathrm{rot}2\pi\right) \tag{4}$$

where $\mathrm{rot} \in \{0, \ldots, \mathrm{nRot}\}$ and $k \in \{1, 2\}$.

3) Performing the first two steps corresponds to creating a single cell in Fig. 1. In order to produce a complete quangle set, the two steps above need to be repeated for $\mathrm{nQuant} = \{2, \ldots, \mathrm{maxQuant}\}$.

To detect objects in a single scale, we use a sliding window approach, where an image is scanned by a so-called search or detection window. Scale invariance is achieved by successively downsizing the original image. In order to look for candidates, these quangle masks need to be assigned to positions (i, j) within the detection window x. This defines, at the same time, our quangle features. We furthermore distinguish between two different types. Equation (5a) describes a quangle feature using the original gradient angle, whereas double angle representation is employed in (5b)

$$q_1(x, i, j, \tau_1, \tau_2) = Q(\tau_1, \tau_2, \angle \nabla x(i, j)) \tag{5a}$$

$$q_2(x, i, j, \tau_1, \tau_2) = Q(\tau_1, \tau_2, \mathrm{mod}(2 \cdot \angle \nabla x(i, j), 2\pi)). \tag{5b}$$

Both quangle feature types in the equations above take the detection window x, the position (i, j) within x, and a particular quangle mask out of $\{Q\}$. Using both q_1 and q_2, the number of possible features is determined by the size of the detection window and the number of quangle masks in $\{Q\}$. We include both single and double angle representation in our set of quangle features since they are meaningful at different sites within the search window. Considering a face, the original gradient is more informative between the landmarks because this information can effectively differentiate between dark-light and light-dark transitions which get lost when doubling the angle, to be discussed next. The double-angle representation, which maps ϕ to 2ϕ has been shown to represent the structure tensor eigenvector directions. Despite being a simple linear mapping, the 2ϕ mapping does not carry the same information as ϕ because ϕ is an angle and the mapping 2ϕ is on the ring of $[0, 2\pi]$ (not on the ring of $[-\infty, \infty]$) which creates an equivalence class [15] for angles that differ from π. Thus, ∇f and $-\nabla f$ are equivalent after the 2ϕ mapping which represents the linear structures more effectively that are invariant to a rotation with π. Such structures include, but are not limited to, lines. The double angle representation at boundaries is thus more resistant to illumination and background changes, because it can represent the symmetry inherent to such points more effectively. Accordingly, both single angle and double angle features are complementary and meaningful features to represent objects/faces.

B. Classifier Building

A good classification (yielding a low error rate) cannot be obtained with a single quangle feature, but obviously, it is neither meaningful nor practical to evaluate all of them within the detection window. In order to find the most suitable quangles, we employ AdaBoost [10], which provides a very efficient feature selection procedure. In the process, a number of good features (called weak classifiers) are combined, yielding a so-called strong classifier. In the following, we provide a brief step-by-step description of how such a strong classifier can be built using AdaBoost learning.

Step 1) x_i denotes the training images and $y_i \in \{0, 1\}$ labels them as positive or negative class examples.

Step 2) Weights' initialization $w_{1,i} = (1/2m), (1/2l)$, where m and l are the number of positive and negative class examples.

Step 3) Weights' normalization $w_{t,i} \leftarrow \left(w_{t,i}/\sum_{j=1}^{n} w_{t,j}\right)$, where n denotes the total number of training images and t (initially 1) is used as the weak classifier index.

Step 4) Weighted error calculation $e_t = \min_{i,j,\tau_1,\tau_2,k} \sum_i w_i \cdot |q_k(x, i, j, \tau_1, \tau_2) - y_i|$ where $k \in \{1, 2\}$ represents the according feature type (single or double angle).

Step 5) Select the weak classifier with the lowest error $h_t(x) = q_{k_t}(x, i_t, j_t, \tau_{1_t}, \tau_{2_t})$, where the parameters i_t, j_t, τ_{1_t}, τ_{2_t}, and k_t minimize the error.

Step 6) Weights' updating $w_{t+1,i} = w_{t,i} \cdot \beta_t^{1-|q_t(x_i)-y_i|}$, where $\beta_t = (e_t/1 - e_t)$.

Step 7) Step 3–6 are repeated T times, which is the desired number of weak classifiers.

Step 8) The final strong classifier

$$C(x) = \begin{cases} 1, & \sum_{t=1}^{T} \alpha_t \cdot h_t(x) \geq \frac{1}{2} \sum_{t=1}^{T} \alpha_t \\ 0, & \text{otherwise} \end{cases} \tag{6}$$

with $\alpha_t = \log(1/\beta_t)$.

Following the description above, we obtain a strong classifier, which is composed of T weak classifiers. Adding more weak classifiers will probably result in a higher detection rate (lower error) but unfortunately also directly affects the computation time, in the operational as well as training phase. Compared to the rectangle features originating from [11], our features omit a parameter yielding a tremendous speedup during training. This will be explained further.

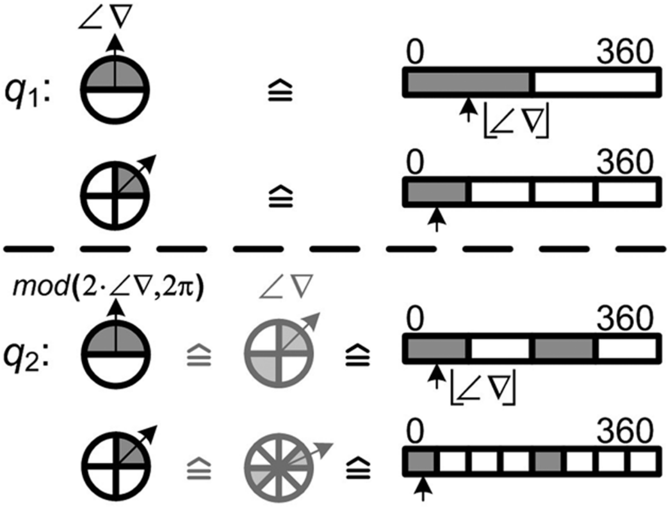

Fig. 2. Example lookup tables (rightmost) representing quangle masks in case of single angle (top) and double angle (bottom) representation. The original gradient angle is floored in $[0,360]$ and used as an index for the binary lookup tables. Furthermore, for the evaluation of any mask used in q_2, a "helper quangle" can be imagined, which directly corresponds to double angle representation, and exists only in the form of lookup tables. This is shown in the lower part of the figure.

An alternative to the single strong classifier is the so-called cascaded classifier scheme, which is computationally efficient and provides high detection rates. The idea is to create a number of less complex strong classifiers and to evaluate them consecutively. A single negative decision at any level of such a cascade leads to an immediate disregard of the concerned candidate image region. In this case, the classifiers at the remaining levels do not need to be evaluated, whereas positive decisions trigger further consideration of other (weak) classifiers. Note that the final detection rate D for the cascade can be expressed in detection rates per level d, such that $D = \prod_{i=1}^{K} d_i$. This is also valid for the final false-positive rate F and the false-positive rate per level f, yielding $F = \prod_{i=1}^{K} f_i$. In both cases, K is the number of levels. In this way, each cascade level has to keep nearly all of the positive training data but on the other hand, only needs to sort out a portion of negative examples in order to deliver good results (e.g., a ten-level cascade with $f = 0.33$ will lead to a false-positive rate of 1.5×10^{-5}. This fact makes it possible to achieve very high detection rates per level, while keeping down the false positives. The configuration of a series of strong classifiers is driven by the detection and false-positive rates per level. First, the detection rate of the current strong classifier at the top layer is checked. If d is too low, the threshold of the classifier [right-hand side of the inequality in (6)] is successively reduced until a predefined d is reached. If f is still obeyed, the level is complete. Otherwise, further weak classifiers have to be added until both rates d and f are complied with.

When establishing the cascade (i.e., configuring and training a strong classifier per level), we apply a bootstrapping strategy. After completing each level, the rejected negative class examples are replaced by new ones, which the cascade would still

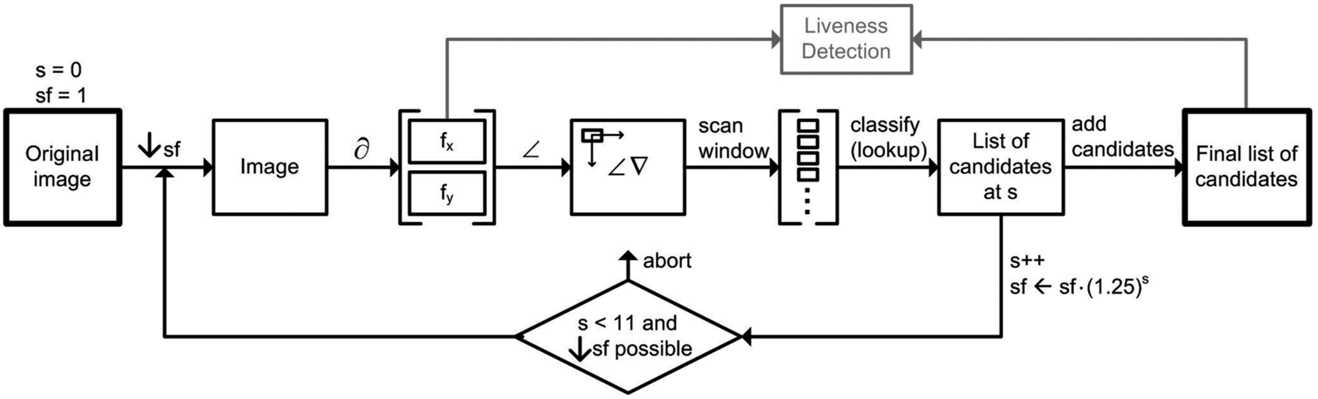

Fig. 3. Object/face detection process. An input image is investigated at 11 scales. At each of them, the angle of the gradient is analyzed by a trained (cascaded) classifier within a sliding window. The object candidates of each scale are integrated in a final list and multiple detections are eliminated.

(wrongly) classify as being positive. It is worth mentioning that this remarkably prolongs the training since it becomes naturally more unlikely to encounter such examples. Nevertheless, this results in a very discriminative cascade, in which the classifiers become more complex, as they focus on trickier examples.

C. Implementation

In order to make the object detection computationally efficient, the number of operations required for each detection window has to be minimized. By means of the cascaded classifier approach described in the previous section, it is possible to make early decisions on whether a detection window is useful or not. This is in favor of the execution time, since only a few features have to be evaluated in case of nonobject sites. Furthermore, we can reduce the number of operations needed to calculate and classify a single feature. In this study, we employ so-called lookup tables to speed up this process. From a programmer's point of view, they are simple arrays containing precalculated values, which are accessed by their indices. Recalling the quangle features of type q_1 and particularly q_2 in (5a) and (5b), lookup tables provide an effective solution for both of them. This is especially beneficial, since we can avoid a separate calculation of the double angle representation in case of q_2 and, thus, save many operations. Fig. 2 depicts two exemplary lookup tables for both q_1 and q_2. Each quangle mask is represented by a binary lookup table, which uses the angle as an index. The value at a certain position corresponds to 1 or 0 depending on whether the index (angle) is within a certain partition or not. This partition is defined by the respective quangle mask. In order to be able to use the angle as an index, we floor the original gradient angle to integer values in [0,360[. However, the quangle features of type q_2, need some further attention. As visualized at the bottom of Fig. 2, $\bmod(2 \cdot \angle\nabla, 2\pi)$ can be represented by the ordinary gradient angle $\angle\nabla$ by means of "helper quangles" (displayed in light gray), only existing in the form of lookup tables. First, we divide the thresholds τ_1 and τ_2 of the original quangle mask by two. The resulting partition, together with a 180° shifted version, is set to 1 in the respective lookup table. Since this corresponds to an inversion of the double angle calculation, we can use the original gradient angle $\angle\nabla$ as an index, yet obtaining a classification for the doubled angle. As a consequence, 1 array access is needed per weak classification for any quangle.

Having a trained (cascaded) classifier, we can use the algorithm illustrated in Fig. 3 to detect objects/faces within any given image. The image to be analyzed serves as a starting point at scale $s = 0$ and scale factor $sf = 1$. In the next steps, we calculate the gradient [see (1)] of the whole image at scale s using separable Gaussians and their derivatives and extract the angle information. After this, we scan the image with the detection window. The trained classifier describes a sequence of thresholds and weights for the most discriminative quangle features to be evaluated. The actual classification is performed using the lookup tables introduced above. Having the candidates of the first scale, we successively reduce the image size by a factor 1.25 and start over with the calculation of the gradient. This is done for ten further scales or until the new size would become smaller than the detection window. The objects detected at each scale are integrated in a final list of candidates. In order to ensure that an actual object is only "counted" once (indicated by a single, framing rectangle in the image), neighboring candidates in position and scale are grouped (averaged).

D. Face and Facial Landmark Detection

In this section, we apply the object detection system introduced above to face and facial landmark detection. The size of the search window for face detection is 22×24 pixels. Our system operates in real time at a resolution of 640×480 using 11 scales on a standard desktop computer. We have been collecting approximately 2000 faces of varying quality from online newspapers for training purposes. In addition, to compensate for the difference in size and rotation, all face images were aligned by means of three facial landmarks, namely the mouth and both eyes. With this, slight rotations in the interval $[-10°, \ldots, +10°]$ are applied to the training faces. Some background is included in a typical positive (face) example. On the other hand, the negative examples are chosen randomly from many images, which do not contain any faces. Also, when training a cascaded classifier, further negative examples are automatically "harvested" from these images.

In the following, we present a small experiment to provide some important initial results and insights. In order to strengthen the argument in Section II-A, where we suggest the use of both

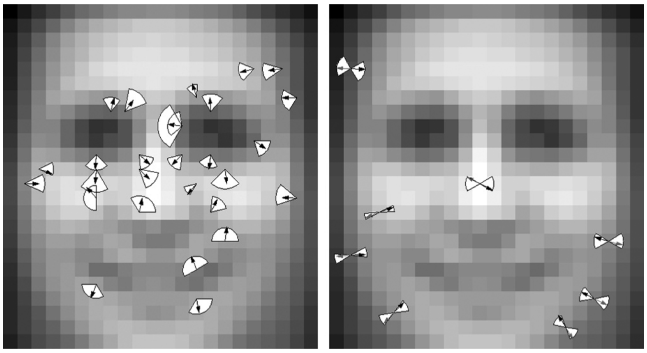

Fig. 4. Exemplary strong classifier employing both single and double angle features, which are displayed side by side. Most of the single angle features (left) can be found in the inner facial region, whereas double angle features (right) are likely to be positioned at the face boundary. The black arrows point in the gradient direction, whereas the gray arrows indicate a 180° shifted version in case of the double angle features.

single and double angle features, we train a strong classifier employing both feature types versus classifiers using either of the features. Empirical tests on a subset of positive examples (900) and 9000 negative examples revealed that $\{Q\}_{8,5}$ is an eligible set of quangle masks for face detection. The least number of quangle features to separate these 9900 examples error free served as a criterion, besides economic parameters for the set $\{Q\}$. By doing so, we also advanced to reduce the complexity of strong classifiers and, consequently, cascaded classifiers. A number of 36 quangle features (28 of type q_1 and eight of type q_2) were selected in the case of using $\{Q\}_{8,5}$. Fig. 4 visualizes the selected features in separate detection windows. In both cases, the black arrows indicate $\angle\nabla$ of the underlying average face at feature sites. The white partitions show the range that the respective angle is supposed to be in. The hourglass-shaped partitions in the second image indicate double angle features, which would also tolerate if the gradient angle pointed in the opposite direction (indicated by the gray arrows). The radii are modulated by α_t, the weights of the corresponding weak classifier, as an indication of the features' relevance. It can be observed that single angle features frequently occur in the inner facial regions, whereas features of the second type are situated in the bounding regions. In a further step, we trained two strong classifiers using the same training setup, yet employing either features of type q_1 or q_2. Error-free separation of the training data involved 49 single angle or 83 double angle features. Compared to the combined setup, these numbers are significantly higher.

Assuming that we have the sizes and positions of all faces in an image, we continue with facial landmark detection. The focus is on the mouth, since it is needed for "liveness" inspection later. However, the presented concept also works for other facial landmarks. For training purposes, we extracted the mouths from 600 out of the 2000 available faces. The size of the mouth regions is 11 × 7 pixels and small rotations in the interval $[-5°, \ldots, +5°]$ are included as well. All other face parts, but also nonface patterns, are used as negative examples. Furthermore, the training proceeds as implemented for face detection.

Having automatically detected faces, among-scale grouping has affected their observed sizes. Therefore, we use the two (nearest) neighboring scales for mouth detection. In each of them, we classify the mouth detection window at five positions, sampled from $N(\mu, \sigma)$. Here, μ is the expected mouth center within the face detection window and σ is set to 1.5. This corresponds to a biased random search in an approximately 5 × 5 neighborhood. In a final step, the mouth candidates of both scales are grouped in order to eliminate multiple detections.

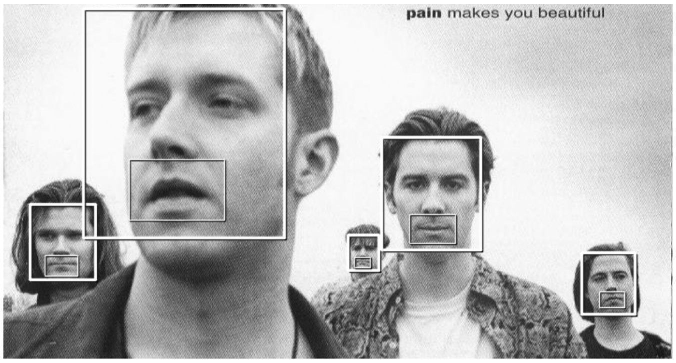

Fig. 5. A (cropped) example image from the CMU–MIT face test set, with faces and mouths detected by the proposed method.

Since we already know the size and the position of each face and do not need to calculate the gradients, face and facial landmark detection can be combined without increasing the computational load noticeably. We show an example of combined face and mouth detection by our method in Fig. 5.

III. "Liveness" Analysis

We stated that our "liveness" verification barriers need reliable face (part) detection and optical-flow estimation. Having explained the former, we shall also summarize the method employed for fast optical-flow calculation first. It is worth noting that because we switch into the "liveness" verification module after having detected any face (and a landmark), the gradient images are readily available for it. This offers, besides the usage of common features for all tasks, a significant reduction of computations when calculating the OFL features discussed next.

A. Summary of the OFL

In [16] and [23], the optical flow of lines (OFL) has been proposed to measure the velocity at which the lines move in image sequences (i.e., the technique approximates the velocity component normal to the line direction—the normal flow). The OFL is a lightweight optical flow alternative to all-embracing, but time-consuming variants of motion estimation. The goal of the OFL is to approximate the two components of the normal velocity vectors of a line in the image plane v_x and v_y. In the following, we denote a Gaussian derivative filter with respect to an arbitrary dimension axis z as h_z, where z can be either x, y, or t, the horizontal, vertical, and time dimension axes, respectively. Given a space-time stack of 3–5 consecutive images $\{\mathrm{Im}\}$, the following steps are performed to obtain v_x.

Step 1) Filter each frame of $\{\mathrm{Im}\}$ with h_x to extract vertical lines, and save the result in a helper stack $\{f\}$.

Step 2) Permutate $\{f\}$ along the y-axis to get all 2-D space-time slices, and save them (absolute values only) in $\{\mathrm{xt}\}$.

Step 3) For each $\{\mathrm{xt}\}$, calculate its gradient by filtering with h_x and h_t, and compile a complex representation having the derivatives with respect to x and t as components. Take the square of the complex slices and save them over the previous versions in $\{\mathrm{xt}\}$.

Step 4) Average each $\{\mathrm{xt}\}$ by use of a Gaussian kernel (with larger σ than for h_z), yielding a set of images with complex values representing the linear symmetry [27] or the eigenvector of the local structure tensor $\{\mathrm{ls}\}$.

Step 5) For every $\{\mathrm{ls}\}$ slice, consider the values at the center t-position only and take $\tan(0.5 \cdot \angle \mathrm{ls})$ at every x-position, yielding a single row per slice. Store each row at the corresponding y-position in the final v_x (the y-position at which the original $\{\mathrm{xt}\}$ slice was taken out of f).

Two constraints are to be considered: First, velocities in v_x are only valid if the corresponding values at the center frame of f are (in magnitude) above a threshold. Second, velocities are only valid if they are below a threshold. To calculate v_y, the algorithm above just needs to be fed with the x/y-transposed space-time stack. For details and results on error quantification of the OFL, we refer to [16] and [23].

Like for the gradient approximation in Section II, separable Gaussians and their derivatives are used here for all filtering tasks, requiring only a few operations per point. More specifically, if the object detection from above precedes, we can plug the gradient components within a region of interest (ROI) into $\{f\}$ and proceed straightly to Step 2) in the OFL algorithm that is shown. Usually, Step 1) would involve the biggest dimensions to deal with. In our analysis toward "liveness" detection, we focus on detected faces and mouths as ROIs. Also, we can jump to scales of interest, since the image gradient for a pyramid of the original image is already estimated.

B. Repelling Spoofing Attacks

One can utilize a simple physical fact when assessing whether a face imaged by the system camera is "live" or if it comes from a photograph, presented to the system in a spoofing attempt. In Fig. 6, two frames cut out of camera footage are displayed. The rectangles indicate the detected faces, whereupon the actual face regions are replaced by the velocity magnitudes $|v| = |(v_x, v_y)^T|$. We may use as few as three consecutive frames to calculate the OFL, at face sites only. On the left-hand side of Fig. 6, we can observe that the velocity values remain rather constant, due to the photograph being plain. In contrast, on the right-hand side, we can observe a higher variation of the velocity. This is due to 3-D face landmarks having different distances and mutual relationships with respect to the camera, thus generating nonuniform motion vectors with a specific mutual dependence over the face region [16]. The latter study compared three parts (face center + 2 sides), using Gabor-based techniques. Such a "liveness" barrier could be efficiently implemented using the proposed object detection.

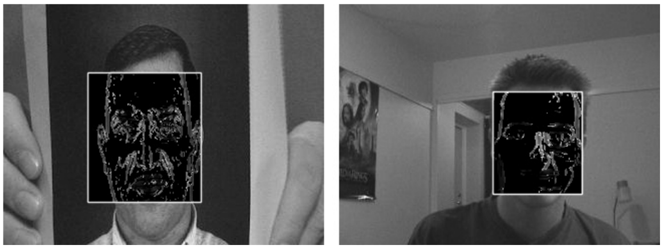

Fig. 6. On the left-hand side, a bent photograph is held in front of the camera, contrary to an actual person being there on the right-hand side. The faces are located (rectangles) by the method from Section II. The OFL is efficiently calculated through three frames in time, and the according (absolute) velocity values are displayed on top. We can observe a larger variation of values in case of the "live" face due to its depth.

However, assuming that a perpetrator manages to place a portable video device that replays a video of another person's face (with expression changes, other movements, speaking, etc.) at an adequate position in front of the system camera, most of the (commercial) recognition systems as well as "liveness" detection modules will have a hard time not to be spoofed. Here, we try to oppose this spoofing attempt in a fully automatic manner, requiring some interaction of the person in question. This interaction occurs through the utterance of a specific digit sequence, either known previously by the person or prompted randomly. The latter strategy is to be favored, since a sound utterance can be easily recorded. We suggest a "liveness" verification barrier functioning autonomously (without assistance from voice) at the visual level, by exploiting how a sequence of digit utterances changes the facial expression. Here, the digits are decoded using the lip motion, on one hand, to enable a comparison to what should have been said, and to be available for comparison on the result of another autonomous (speech) expert's digit recognition of the other. The problem to solve in our approach is to identify digits $0 \ldots 9$, one by one in a sequence, by assigning the observed lip motion to one of the ten classes. First, however, we have to confine the mouth region of a "talking face." In [23], the mouth center was pinpointed manually, to be able to assess the potential of the used motion estimation (OFL). Here, we can employ the face (part) detection from Section II to automatically locate the mouth and proceed through fully automatic lipreading of digits. Thus, the OFL computations are carried out only within an automatically placed ROI centered at the mouth. A square region is used as a sliding window to span space-time stacks of five consecutive frames each. Next, a dimension reduction for the purpose of classification is performed. The calculated velocity vectors $v = (v_x, v_y)^T$ are first reduced to 1-D by projection onto an intuitive stick-mouth model. This is also illustrated on the left-hand side of Fig. 7, which indicates a mouth region divided (by the dashed lines) into six parts. The expected orientations of motion are $-45°$, 90°, and 45° in the upper three regions, and in the reversed order for the lower three parts, also marked by the solid bars in the figure. The velocity vectors in each part are projected onto the corresponding direction, yielding signed scalars. These scalars are further clustered within four parts of the mouth region. This is visualized on the right-hand side of Fig. 7, where dashed lines indicate the consideredfour parts.

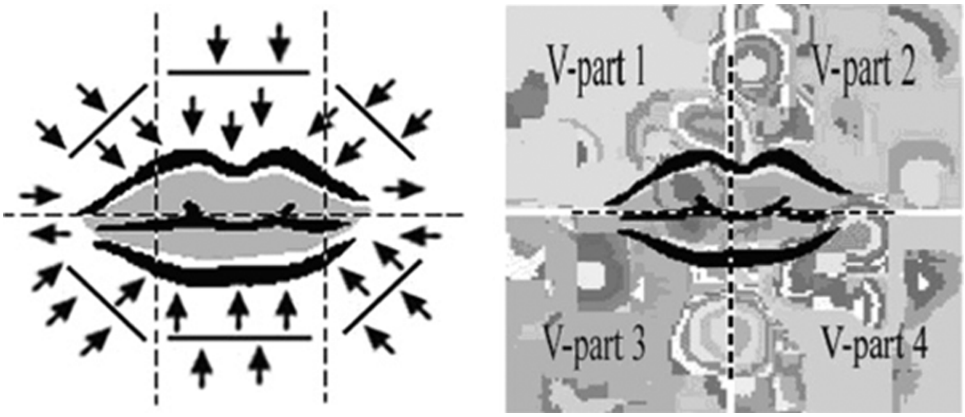

Fig. 7. Dimension reduction of the extracted velocities in the mouth region: On the left-hand side, the 2-D vectors in each part (divided by the dashed lines) are reduced to scalars, by projecting them on the orientations indicated by the solid bars. On the right-hand side, these scalars are clustered within four parts (divided by the dashed lines), with the gray values representing velocity magnitudes.

In each of them, 20 cluster centers (with population 20) are automatically established using fuzzy C-means [28]. The final lip motion statistics are then represented by a 40-D vector per part containing the cluster centers and populations. These 160 dimensional vectors are extracted along a "talking face" video, for a person uttering a single digit. This way, a final matrix g of the size 160 × N, where N is the number of frames, is gathered. For classification, g is reshaped into a row vector used for training a 10-class SVM and to be classified by the ready expert, respectively. LIBSVM [29] was employed for this purpose.

TABLE I
DETECTION RATES AND THE NUMBER OF FALSE POSITIVES ON THE YALE FACE TEST SET

Method	Detection Rate	False Positives
Nguyen [9]	86,6%	0
Proposed method	100%	0

TABLE II
DETECTION AND FALSE POSITIVE RATES ON THE CMU–MIT FRONTAL FACE TEST SET

Method	Detection Rate	False Positive Rate
Rowley [5]	89,2%	$1,27 \times 10^{-6}$
Viola&Jones [11]	92,9%	$1,27 \times 10^{-6}$
Proposed method	94,2%	$1,25 \times 10^{-6}$
Proposed method	93%	1×10^{-6}

IV. EXPERIMENTS

A. Face Detection

For the experiments, the face detector was configured as follows: The size chosen for the detection window was 22 × 24 and the employed quangle masks were in $\{Q\}_{8,5}$. A cascaded classifier, comprised of 22 levels, was trained on 2000 faces. This number suffices since all of the latter belong to different people and include slight, artificially introduced rotations. Each of the cascade's layers had to reject 2000 out of 4000 nonfaces. The total number of weak classifiers in the cascade was 700. Such classifier complexity is very small compared to a couple thousand as suggested in [11] and [13]. This is due to the quangles employing derivative features and, in effect, incorporating more discriminative information than rectangle features. In operation, the first two levels of the cascade, comprising only three and five quangle features, respectively, are already able to reject 75% of all nonfaces. Furthermore, we used $s = 0$ (original resolution) as the starting scale and $\mathrm{sf} = 1.1$ as the factor for downsizing.

The performance of this face detection system is benchmarked on two publicly available databases, namely, the YALE [25] and the CMU–MIT [5] face test sets. Extreme illumination changes are the main challenge of the former test set, which consists of 165 frontal face images of 15 subjects. Especially, the changing light sources, often resulting in semi-illuminated faces, present a major source of errors/false negatives. In addition, facial expressions are varied (sad, sleepy, surprised, etc.). The background is monotonous, apart from faces' shadows cast on the wall, and there is only little variation in the scale of the faces. Table I shows the detection rates and the number of false positives of our method together with the ones for the face detection algorithm proposed in [9], on the YALE face test set.

The CMU–MIT frontal face test set is among the most commonly used data sets for performance assessment of face detection systems. It is composed of 130 images containing 507 frontal faces in total. The quality of the images varies substantially, besides the large variation in the scale of the faces which increases the difficulty of the detection task. Even a number of simple, hand-drawn face examples occurs. In addition to the detection rate, this set also permits representative numbers for the false positives because it contains many high-resolution images. In Table II, the results of our technique on the CMU–MIT frontal test set are related to those of two prominent face detectors [5], [11], by adjusting the false positive rate to a common level. Also, the detection rate achieved by our method at 1 false detection per million evaluated windows is given, constituting our best result.

The results on the YALE test set confirm that our face detection method is resistant to substantial illumination changes without performing any (histogram related) preprocessing. Note that the latter is actually done in all methods we compare our results to. In Fig. 8, two "YALE faces" are shown, with indicated detections by the proposed method. Note the severity of the illumination conditions. Moreover, regarding the achievements on

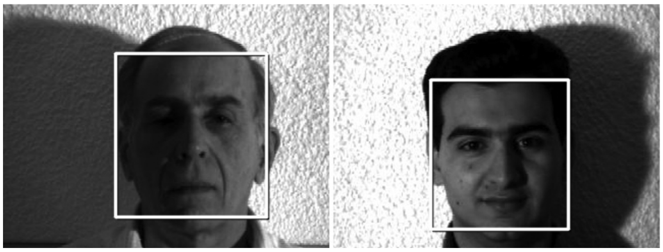

Fig. 8. Two images from the YALE face test set, illustrating what we consider "severe" illumination changes, managed by the proposed method though, as can be seen.

the CMU-MIT test set, the proposed technique reveals highly competitive performance.

B. "Liveness" Verification

In order to explore the potential of the proposed "liveness" verification barrier with a digit-prompted system, we use a number of 100 users from the XM2VTS audio-visual database [26]. For each user, a "talking face" video in which the person's face was recorded while speaking aloud, the sequence "0 1 2 3 4 5 6 7 8 9" is available. The main goal here is to recognize the digits by lip-motion only, to assess its value for "liveness" independent of other modalities. For test purposes, these videos were segmented semiautomatically so that short image sequences of single-digit utterances were available in the end, yielding 100 short videos for every digit. Furthermore, the dataset was divided into 60% training and 40% test set. We used the public SVM implementation of [29] for the experiment. Presented results were attained by use of an RBF-kernel, adjusted with the help of cross validation over the training set. During the experiment, the velocity feature vectors were extracted at the mouth region and given to the 10-class SVM for each single-digit video.

For automatically locating the mouth, we employ the cascaded classifier from the experiments above plus an ad-hoc cascaded classifier for mouth detection only. Using the face–mouth combination shortens the search for the mouth region within a video frame, also allowing for reduced training efforts for the mouth detector. The latter uses quangle masks from the set $\{Q\}_{8,5}$ within the features as well, and has a complexity of 310 weak classifiers (spread out to 14 layers). This relatively large number of utilized quangle features comes presumingly from the characteristics of the mouth region, not being as discriminative as, for example, the upper face part or the whole face. Furthermore, no emphasis was put on training on especially opened mouths (as during speaking). The size of the ROI for optical-flow extraction was kept constant at 128 × 128 pixels, which was possible due to negligible face/mouth scale variations throughout the videos.

TABLE III
CONFUSION MATRIX FOR THE RECOGNITION OF DIGITS "0" TO "9," UTTERED BY PERSON "005" OF THE XM2VTS DATABASE, WHEN AUTOMATICALLY EXTRACTED LIP-MOTION IS CLASSIFIED BY AN SVM EXPERT

	0	1	2	3	4	5	6	7	8	9
0	0,8	0	0	0	0	0	0	0	0	0,2
1	0	0,6	0	0	0	0	0	0	0	0,4
2	0,4	0	0,6	0	0	0	0	0	0	0
3	0,3	0	0	0,7	0	0	0	0	0	0
4	0,3	0	0	0	0,7	0	0	0	0	0
5	0,3	0	0	0	0	0,7	0	0	0	0
6	0,3	0	0	0	0	0	0,7	0	0	0
7	0,3	0	0	0	0	0	0	0,7	0	0
8	0,2	0	0	0	0	0	0	0	0,8	0
9	0	0	0	0	0	0	0	0	0,2	0,8

In Table III, we show the confusion matrix obtained for person "005" for all digits "0" to "9," containing the probability estimations of (false) assignment of the automatically extracted lip-velocities by the SVM. Looking at it, for example, the probability of classifying an uttered digit as "9," when "0" would be correct is 0.2. Accordingly, the probability of falsely identifying an uttered digit is the sum of all nondiagonal elements of the concerned row (e.g., $p\left(1|1^C\right) = (1 - p_1) = 1 - 0.6 = 0.4$ for digit "1"). Aggregating over all digits, the average misclassification probability $\tilde{p}$ for a particular person is expressed

$$\tilde{p} = \frac{1}{10} \cdot \sum_{i}^{10} p\left(i|i^C\right). \tag{7}$$

Successful digit recognition for exemplary person "005" involving all utterances is accomplished with a probability of approximately 0.7, with $\tilde{p}$ being 0.3. Furthermore, one could exclude the most frequently confused digits (e.g., "0") to support the classification. Successful recognition of single-digit utterances for all 100 persons, based on an averaged $1 - \tilde{p}$ is accomplished with probability 0.73. This should be viewed as an indication for the potential of using bare lip-motion to assess "liveness."

Assuming an error-free lip-expert for digit recognition, any disagreement between uttered and the expected sequence (known by the user or randomly prompted) can, of course, be considered a spoofing attempt. Nevertheless, the improbability of an actual attack using the correct sequence also enables a moderate machine expert to be highly useful. It is worth pointing out the difficulty of performing text-prompted speech recognition only from video, not the least experimentally, because the amount of data to be collected is an order of magnitude larger than for speech.

Finally, we will give the results for face/mouth detection on the 100 used videos from the XM2VTS database, which has controlled conditions (uniform background and high resolution). Our face detector was successful in all observed frames with no false positives. We also quantify the accuracy of the mouth detection/localization. From previous studies [23], we have manually pinpointed (center) coordinates of the mouth for the 100 original videos (only for the first frame though) available. We compare these with the automatically derived ones for the same frames. In Fig. 9, the differences that occurred in x and y-values are displayed on the left- and right-hand side, with the corresponding means and standard deviations being -1.7 ± 3.5 and -1 ± 3.1, respectively. The systematic deviation in mean, which is a matter of detector bias, was compensated for in the experiment shown. Note that the standard deviations of 3.5 and 3.1 are to be seen in context of a 720 × 576 frame, with a (roughly) estimated average face and mouth size of 280 × 300 and 80 × 50 pixels to be located, respectively. Taking also into account that an image-based

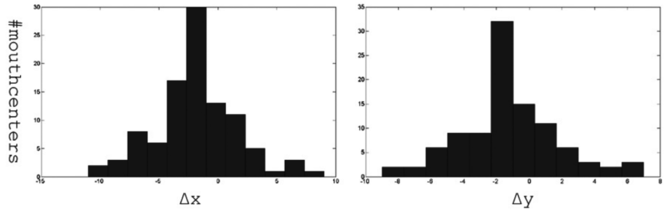

Fig. 9. Histograms showing the differences in x- and y-coordinates on the left- and right-hand side, respectively, when comparing manually with automatically pinpointed mouth centers for the initial frames of 100 "talking face" videos. Note that the average mouth size is roughly 80 × 50 pixels.

method (in contrast to landmark-based) was used, this accuracy is satisfying.

V. Discussion

In this section, we tie the facts that differentiate the proposed methods against related approaches.

An important factor for the performance of an object detector is speed, not excluding training time, since fast training will enable a system to rapidly adapt to new tasks. Viola and Jones, for example, reported in [11] that the training time of their final classifier was in the order of weeks on a single machine. Employing our features, the training of a comparable cascade takes about an hour on an ordinary desktop computer. To be fair with respect to computational power advancements, we have timed the training of the 900/9000 classifier from Section II-D for both of nour features and Viola and Jones' using OpenCV. The latter provides functionality that directly implements Viola and Jones' approach, presumingly with high efficiency. A standard desktop computer with a 2.2-GHz Intel processor and 1 GB of random-access memory (RAM) was used for this purpose. It turned out that our (error-free) 36-quangle classifier was ready after 17 min. On the contrary, the same training lasted 529 min and resulted in a 133-feature classifier using Viola and Jones' original basis functions. Since we can use less features, which do not require the greedy search for the best thresholds on the training data, this translates accordingly to the training/complexity of a detection cascade. We infer that rectangle features, which rely on gray-level information, are not as discriminative as our quantized gradient-angle features and need to be calibrated more precisely. Furthermore, the operational speed of the proposed scheme (our implementation) is compared with the object detector's included in OpenCV (Viola and Jones based). Compared to the improvement in training speed, the face detection speed does not differ remarkably which is probably related to our implementation. To process a 1280 × 1024 image on the mentioned system (standard desktop), both methods take about 2 s.

Other studies have suggested schemes for reducing classifier complexity [13], [14], which we did not investigate yet, because our combined (q_1 and q_2) features resulted in a classifier that is simple enough. In a related study [9], using a naive Bayes classifier, the original gradient angle was merely quantized by a fixed number of seven steps without a further study of flexible and lower quantization levels. In [12], no quantization at all was done (except for integer conversion) and only the structure tensor directions (the doubled gradient angle) were used. Furthermore, the weak classifiers were constructed differently, involving significantly more operations for evaluation. Also, most object detection methods depend on a preprocessing step (gray value normalization, equalization) prior to feature extraction, which we can omit because we exclusively operate on the gradient angle.

Coming to the application part, other studies have suggested tracking mouth minutiae (e.g., corner points [9] or lip contours [22]) (not for "liveness" purposes though). In a real environment, these approaches may suffer heavily from noise. Another disadvantage is the nonconstant computation time due to the iterative process in convergence of the contour fitting. Instead, we have (both the face and) the mouth robustly located by the introduced method frame by frame, and we reuse the gradient to calculate the OFL within the mouth region in real time. First, this assistance equates to savings in computations (e.g., for gradient approximation and interpolation). Had we used an object detection scheme employing different features, we would have had to compute them increasing the processing load. Second, the chances of overcoming severe noise are great, since the object detector has already been trained on numerous realistic samples. Finally, based on our knowledge, no work has been presented on digit recognition based on visual observations only. Our motion estimation technique can be used to separate the pauses from speech [23], but there are more experiments to be conducted with respect to automatic digit segmentation in sequences. In its current state, the system is round-driven with a single digit per time slot.

VI. Conclusion

In this study, we presented a novel real-time method for face detection. However, it is possible to use the technique as a general image-object detector. An experimental support of this view is its straightforward usability to detect mouths. The introduced quantized angle ("quangle") features were studied experimentally and we presented evidence for their richness of information measured by their discriminative properties and their resilience to the impacts of severe illumination changes. They do not need preprocessing (e.g., histogram equalization/normalization), adding to their computational advantage. This was achieved by considering both the gradient direction and orientation, yet ignoring the magnitude. A quantization scheme was presented to reduce the feature space prior to boosting [i.e., it enables fast evaluation (1 array access)]. Scale invariance was implemented through an image pyramid. The training excels in rapidness, which enables the use of our object detector for changing environments and application needs. Furthermore, we proposed novel strategies to avert advanced spoofing attempts such as replayed videos (and presented photographs), which took advantage of calculations from the object detection stage. For this purpose, we showed the possibility of restoring utterances of a person by analyzing the motion of the lips only. The practicability of the proposed methods and ideas was corroborated by satisfying experimental results for face detection (e.g., 93% detection rate at a 1×10^{-6} false positive rate on the CMU–MIT frontal face test set) and mouth detection as well as "liveness" assessment.

References

[1] M.-H. Yang, D. Kriegman, and N. Ahuja, "Detecting faces in images: A survey," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 1, pp. 34–58, Jan. 2002.

[2] M. Hamouz, J. Kittler, J. Kamarainen, P. Paalanen, H. Kalviainen, and J. Matas, "Feature-based affine-invariant localization of faces," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 9, pp. 1490–1495, Sep. 2005.

[3] M. Turk and A. Pentland, "Eigenfaces for recognition," *J. Cognit. Neurosci. (Winter)*, vol. 3, no. 1, pp. 71–86, 1991.

[4] K. K. Sung and T. Poggio, "Example based learning for view-based human face detection," Cambridge, MA, 1994, Tech. Rep.

[5] H. Rowley, S. Baluja, and T. Kanade, "Human face detection in visual scenes," *Proc. Advances in Neural Information Processing Systems 8*, pp. 875–881, 1996.

[6] C. Cortes and V. Vapnik, "Support-vector networks," *Mach. Learning*, vol. 20, pp. 273–297, 1995.

[7] E. Osuna, R. Freund, and F. Girosi, "Training support vector machines: An application to face detection," in *IEEE Conf. Computer Vision and Pattern Recognition*, San Juan, Puerto Rico, Jun. 17–19, 1997, pp. 130–136.

[8] H. Schneiderman and T. Kanade, "Probabilistic modeling of local appearance and spatial relationships for object recognition," in *Proc. CVPR*, 1998, vol. 00, pp. 45–45.

[9] D. Nguyen, D. Halupka, P. Aarabi, and A. Sheikholeslami, "Real-time face detection and lip feature extraction using field-programmable gate arrays," *IEEE Trans. Syst. Man Cybern. B, Cybern.*, vol. 36, no. 4, pp. 902–912, Aug. 2006.

[10] Y. Freund and R. E. Shapire, "A decision-theoretic generalization of online learning and an application to boosting," *J. Comput. Syst. Sci.*, vol. 5, no. 1, pp. 119–139, 1997.

[11] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Proc. CVPR*, Kauai, HI, Dec. 2001, pp. 511–518.

[12] B. Froeba and C. Kueblbeck, "Real-time face detection using edge-orientation matching," in *Proc. 3rd Int. Conf. Audio- and Video-Based Biometric Person Authentication*, London, U.K., 2001, pp. 78–83.

[13] S. Z. Li, L. Zhu, Z. Zhang, A. Blake, H. Zhang, and H. Shum, "Statistical learning of multi-view face detection," in *Proc. 7th Eur. Conf. Computer Vision-Part IV*, London, U.K., 2002, pp. 67–81.

[14] J. Sochman and J. Matas, "WaldBoost learning for time constrained sequential detection," in *Proc. IEEE Comput. Soc. Conf. Computer Vision and Pattern Recognition*, Washington, DC, 2005, vol. 2, pp. 150–156.

[15] J. Bigun*, Vision with Direction*. Berlin, Germany: Springer, 2006.

[16] K. Kollreider, H. Fronthaler, and J. Bigun, "Evaluating liveness by face images and the structure tensor," in *Proc. 4th IEEE Workshop on Automatic Identification Advanced Technologies AutoI*, Buffalo, NY, Oct. 17–18, 2005, pp. 75–80.

[17] F. Smeraldi and J. Bigun, "Retinal vision applied to facial features detection and face authentication," *Pattern Recognit. Lett.*, vol. 23, pp. 463–475, 2002.

[18] J. Bigun, H. Fronthaler, and K. Kollreider, "Assuring liveness in biometric identity authentication by real-time face tracking," in *Proc. IEEE Int. Conf. Computational Intelligence for Homeland Security and Personal Safety*, Venice, Italy, Jul. 21–22, 2004, pp. 104–112, IEEE Catalog EX815, ISBN 0-7803-8381-8.

[19] J. Li, Y. Wang, T. Tan, and A. K. Jain, "Live face detection based on the analysis of fourier spectra," in *Biometric Technology for Human Identification*. Bellingham, WA: SPIE, 2004, pp. 296–303.

[20] T. Choudhury, B. Clarkson, T. Jebara, and A. Pentland, "Multimodal person recognition using unconstrained audio and video," in *Proc. 2nd Int. Conf. Audio-Visual Biometric Person Authentication*, Washington, DC, Mar. 22–23, 1999.

[21] I. H. W. G. C. H. Bredin and A. Miguel, "Detecting replay attacks in audiovisual identity verification," in *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing*, Toulouse, France, May 14–19, 2006.

[22] S. Dupont and J. Luettin, "Audio-visual speech modelling for continuous speech recognition," *IEEE Trans. Multimedia*, vol. 2, no. 3, pp. 141–151, Sep. 2000.

[23] M. I. Faraj and J. Bigun, "Person verification by lip-motion," in *Computer Vision and Pattern Recognition Workshop on Biometrics*, Piscataway, NJ, Jun. 2006, pp. 37–44.

[24] N. T. J. Luettin and S. Beet, "Speaker identification by lipreading," in *Proc. 4th Int. Conf. Spoken Language Processing*, Oct. 1996, vol. 1, pp. 62–65.

[25] P. N. Belhumeur, J. P. Hespanha, and D. J. Kriegman, "Eigenfaces vs. Fisherfaces: Recognition using class specific linear projection," *IEEE Trans. Pattern Anal Mach. Intell.*, vol. 19, no. 7, pp. 711–720, Jul. 1997.

[26] K. Messer, J. Matas, J. Kittler, J. Luettin, and G. Maitre, "XM2VTSDB: The extended M2VTS database," in *Proc. Audio and Video Based Person Authentication*, 1999, pp. 72–77.

[27] J. Bigun, T. Bigun, and K. Nilsson, "Recognition by symmetry derivatives and the generalized structure tensor," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 12, pp. 1590–1605, Dec. 2004.

[28] J. B. N. Pal, "On cluster validity for the fuzzy c-means model," *IEEE Trans. Fuzzy Syst.*, vol. 3, no. 3, pp. 370–379, Aug. 1995.

[29] C.-C. Chang and C.-J. Lin, "LIBSVM: A library for support vector machines," 2001 [Online]. Available: http://www.csie.ntu.edu.tw/~cjlin/libsvm.

Klaus Kollreider received the M.S. degree in computer systems engineering from Halmstad University, Halmstad, Sweden, in 2004.

His research interests include signal analysis and computer vision, in particular, face biometrics and antispoofing measures by object detection and tracking. He is involved in a European project focused on biometrics BioSecure, where he has also contributed to a reference system for fingerprint matching.

Hartwig Fronthaler received the M.Sc. degree from Halmstad University, Halmstad, Sweden.

His specialization has been in the field of image analysis with a focus on biometrics. He joined the signal analysis group of Halmstad University in 2004. His research interests are in the field of fingerprint processing, including automatic quality assessment and feature extraction. He has been involved in research on face biometrics.

Maycel Isaac Faraj received the M.S. degree in computer systems engineering from Halmstad University, where he is currently pursuing the Ph.D. degree within the signal analysis group.

After contributing to a software product in the field of medical images analysis, he joined the Intelligent System Laboratory at Halmstad University in 2004. His research interests include signal analysis for audio–visual biometrics as well as computer vision and graphics.

Josef Bigun (F'03) received the M.S. and Ph.D. degrees from Linkoeping University, Linkoeping, Sweden, in 1983 and 1988, respectively.

From 1988 to 1998, he was with the Swiss Federal Institute of Technology in Lausanne (EPFL), Lausanne, Switzerland. His research interests include a broad field in computer vision, texture and motion analysis, biometrics, and the understanding of biological-recognition mechanisms.

Dr. Bigun was elected Professor to the Signal Analysis Chair at Halmstad University and Chalmers University of Technology, Gothenburg, Sweden, in 1998. He has co-chaired several international conferences and has contributed as a referee or as an editorial board member of journals including PRL and *IEEE Image Processing*. He served in the executive committees of several associations including IAPR. He has been elected Fellow of IAPR.