FAST EYE LOCALIZATION WITHOUT A FACE MODEL USING INNER PRODUCT DETECTORS

Gabriel M. Araujo[*], *Felipe M. L. Ribeiro,*
Eduardo A. B. Silva,

Universidade Federal do Rio de Janeiro
Cx.P. 68504, Rio de Janeiro, RJ
CEP 21945-970, Brazil
{gabriel.araujo, felipe.ribeiro, eduardo}@smt.ufrj.br

Siome K. Goldenstein

Universidade Estadual de Campinas
Cx.P. 6176
Campinas-SP, CEP 13084-971, Brazil
siome@ic.unicamp.br

ABSTRACT

Many applications in the area of human-machine interaction require accurate and fast facial landmarks localization. Non-invasive techniques are even more essential when the goal is social inclusion. In this work, we propose a novel approach for the detection of landmarks on faces. We introduce a new detector, the *Inner Product Detector* (IPD), based on correlation filters. The main advantages of the proposed method are the tolerance to small variance on the desired patterns, a low computational cost, and the generalization for other features. We evaluate our method using the BioID dataset and compare it to the state-of-the-art techniques.

Index Terms— Eye localization, Facial features, Landmark localization, Object detection, Pattern recognition

1. INTRODUCTION

Facial landmarks can be defined as salient points on images of faces. The problem of locating and/or tracking these spots has been investigated in recent years. These landmarks are useful in many applications, such as biometrics, expression recognition, face alignment, pose estimation, 3D face modeling, and face tracking. Landmarks on eyes, in particular, can estimate attention and gaze of disabled people, in order to control assistive devices.

There are three main ways to locating/detecting eyes [1]: *electro oculography*; *scleral contact lens/search coil*; and *photo/video oculography*. In electro oculography, the position of the eye is obtained by the difference of the electric potential of the skin around the ocular cavity. In scleral contact lens/search coil this position is obtained using a mechanical reference on a contact lens. These two methods are too invasive and require expensive sensors [2].

Photo/video oculography is the least intrusive technique. Some techniques use active infrared (IR) illuminations with good accuracy, but they can not be used in daylight applications and require dedicated hardware [3].

Some techniques are model-based and do not perform any type of learning [3, 4, 5, 6, 7], but that also means they can not be easily generalized to detect other types of features/patterns. These methods use isophote curvature [3], gradients and dot products [4], an adaptive CDF analysis to locate the pupils [5], eye area edge map comparison [6], and projection functions [7].

In contrast, there are several learning-based methods to perform the same task [2, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19]. In [8], for example, a Bayesian method is employed in the learning of a statistical model for the eyes using visual features. A 2-D cascaded AdaBoost in [13]. In [16], the result of a multistage approach with Pairwise Reinforcement of Feature Responses (PRFR) is used as initialization of the Active Appearance Model (AAM). In [11] projection functions are also used, but together with SVM. A scheme with Gabor filters to select features and SVM to classification is used in [15]. In [18] an enhanced version of Reisfelds generalized symmetry transform is employed. In [12] The detection is done trough a scheme using Haar wavelets and SVM. In [20], is used a Bayesian model that combines local detectors (SVM regressors with greyscale SIFT descriptors as features) output with a consensus of non-parametric global models for part locations, computed from exemplars. A codebook of invariant local features is proposed by [19] to represent the eye patterns and a sparse representation classifier with a pyramid-like strategy is used to locate the eyes.

Correlation filtering [21] is a signal processing template-based detection technique for unknown signals. Recognizing this filter as a detector and the unknown signal as a sample to be evaluated, we can use correlation filters in pattern recognition problems such as objects [22], eyes [23] or faces detection [24, 25].

Here, we introduce the *Inner Product Detector* (IPD), a novel approach that employs correlation filters to detect the center of the eye. Since the IPD is a *weak classifier*, we combine it into a cascade structure to obtain a *strong classifier*. Our approach has four distinctive features: (1) The detector inherits the tolerance to small variations of the desired patterns from correlation filters, so it is very accurate. (2) The method is very fast, since it is basically dot products. (3) It is easy to parallelize it for multiple simultaneous features, using

[*]The author is also with Centro Federal de Educação Tecnológica (CEFET/RJ–UnED-NI). Address: Estrada de Adrianópolis, 1.317 – CEP: 26041-271, Nova Iguaçu – RJ – Brazil

This work was partially supported by HP Brazil R&D and Brazilian agency CNPq.

threads or GPU. (4) It is a flexible framework – we can train it to detect any type of objects and landmarks.

2. INNER PRODUCT DETECTOR

Correlation filtering calculates the cross correlation between the filter and an unknown signal [21]. The advantage of this technique is the tolerance to small variations of the pattern to be detected. Correlation filtering has been used in vision, for example, in face detection [24, 25].

Our method uses a new detector based on correlation filters, the IPD (Inner Product Detector). It inherits the tolerance to small variations from correlation filters, but has the advantage to easily incorporate the properties of the problem into the design of the detector. The main differences between our method and traditional dot product (also known as Correlation Filter (CF)) are: (i) CF performs a cross correlation in the frequency domain, while IPD does it in the pixel domain. (ii) CF can only incorporate statistics by weighting the samples DFT, but IPD naturally incorporates a priori statistics in the autocovariance matrices.

Let $\mathbf{X}$ be a d-dimensional random variable with a realization $\mathbf{x}$. This realization can be associated with one of two classes, C_1 or C_0. We aim to find a d-dimensional detector h_{C_1}, optimal in the least square sense, capable of detecting an object that belongs to C_1. Ideally, we want the following classification rule:

$$\mathbf{h}_{C_1}^{\mathrm{T}}\mathbf{x} = \begin{cases} 1, \text{ if } \mathbf{x} \in C_1 \\ 0, \text{ otherwise.} \end{cases} \tag{1}$$

With $[\cdot]^{\mathrm{T}}$ denoting transposition. The least square solution for $\mathbf{h}_{C_1}$ is [26]

$$\mathbf{h}_{C_1} = \left(\sum_{j=0}^{1} p(C_j)\mathbf{R}_{C_j} \right)^{-1} p(C_i)\boldsymbol{\mu}_{C_i}. \tag{2}$$

Where $\mathbf{R}_{C_j}$ is the autocorrelation matrix of the training samples from C_j, $\boldsymbol{\mu}_{C_j}$ is the respective mean, and $p(C_j)$ is the probability of a sample being from C_j. The expression $\sum p(C_i)\mathbf{R}_{C_j}$ is a weighted sum of the correlation matrices of all classes. Again, in this case, the weights are the probabilities of the classes. This term should be invertible, so the number of different samples should be greater than the size of the vectors. This is convenient when the features do not have large dimensions, as in the case of landmarks. In the expression $p(C_i)\boldsymbol{\mu}_{C_i}$, the mean of the desired class is weighted by its probability.

2.1. Landmark Localization Using IPD

From a practical point of view, we have a strong constraint in the IPD's formulation. As can be seen in Equation (1), we suppose the classes are orthogonal, an unlikely assumption in real world data. This means that negative samples might have an IPD value greater than positive samples. We can also have IPD values outside the interval $[0, 1]$. We can use the cosine of the angle between the detector and the sample to avoid this issue, but even in this situation, the IPD value would lie in the interval $[-1, 1]$. One could set up a threshold to select an interval of the IPD value that correspond to some percentage of the training samples. It is particularly useful when using the IPD as a previous step for another method [27]. Even so, the output of the classifier is a cloud of points, but with high probability of being the desirable feature. In order to obtain a single location as result, we simply pick up the output point $\mathbf{y}_{out}$ with bigger IPD value

$$\mathbf{y}_{out} = \underset{\mathbf{y}\in\mathbf{C}_{out}}{\arg\max} \left(\frac{\mathbf{h}_{C_1}^{T}\mathbf{y}_n}{\|\mathbf{h}_{C_1}\|\|\mathbf{y}_n\|} \right), \tag{3}$$

where $\mathbf{C}_{out}$ is a set with N test samples $\mathbf{y}_n$.

3. EXPERIMENTS AND DISCUSSION

In order to provide statistical significance to our results, we use a *10-fold* cross validation in all our experiments (152 images per fold). All results presented in this section are the average throughout the folds.

The IPD works as a binary classifier. Each sample is a block centered in a point inside the ROI (Section 3.2). For training, the positive examples are the blocks whose central point is the coordinate of the ground truth (manual annotation) and its 8-connected neighbors – that provides nine positive samples per image. The negative class consists of all the other points inside the ROI that do not belong to the positive class. During both training and testing, we subtract the average of all pre-processed blocks.

3.1. Dataset

In this paper's experiments we use the BioID dataset [28]. It consists of 1521 gray level images of 23 subjects in frontal pose with a resolution of 384×286 pixels. This dataset is provided with two sets of annotations. The first set describes the coordinates of 20 landmarks in each face. The second set of annotations lists the coordinates of the center of the eyes.

This dataset contains images of individuals wearing glasses and with closed eyes. The illumination and background can have large variations, and the faces appear with different scales and rotations. The capability to deal with theses issues indicate that the proposed method is robust to small variations of the desired pattern, an inherited characteristic from correlation filters.

3.2. Pre-processing

In this work, before the training and testing, we pre-process the images from BioID. The first step is the localization of the face, and we use the OpenCV implementation of the Viola-Jones [29] detector. After that, all the faces are rescaled to 100×100 pixels, in order to reduce the effect of scale change. More precisely, in our methodology, we rescale the detected faces to a common size, independently of the original image or detector size, and perform localization using a sliding window with resolution of 27×27 pixels.

Although happening only in a small number of cases, there are images in which the Viola-Jones does not find the face, and we do not consider such images when evaluating our method. On the average, the Viola-Jones missed 5.3 of the 152.1 images per fold.

To improve robustness to illumination changes and/or non linear illumination we normalize the illumination [30]. This technique consists in a sequence of steps such as: gamma correction, Difference of Gaussians filtering (DoG filtering), and contrast normalization. We used the parameters values recommended by [30] in each step.

Since human faces have a strong prior, with vertical symmetry and similar eye location, there is no reason to search for every interest points in all the face.

We search the center of the eyes in a *Region of Interest* (ROI) with high probability of containing the desired point. We learn this region from the training set – the ground truth provided with the dataset. We assume a Gaussian distribution with mean $\boldsymbol{\mu}_{\mathcal{X}}$ and covariance $\boldsymbol{\Sigma}_{\mathcal{X}}$, where $\mathcal{X}$ is a random variable with N realizations equal the number of manual labels in a fold. For each eye, we take the annotated point $\mathbf{x}_{\max}$ of the training set that maximizes the Mahalanobis distance d:

$$d = \sqrt{(\mathbf{x} - \boldsymbol{\mu}_{\mathcal{X}})^t \, \boldsymbol{\Sigma}_{\mathcal{X}}^{-1} \, (\mathbf{x} - \boldsymbol{\mu}_{\mathcal{X}})} \qquad (4)$$

The ROI is determined applying a tolerance of 5% to the maximum Mahalanobis distance, $d_{\max}$. This way, we select a region of the image with a high probability of containing the eye's center. To evaluate if a candidate point $\mathbf{x}_c$ belongs to the ROI, it suffices to verify if the condition of the Equation (5) is satisfied.

$$(\mathbf{x}_c - \boldsymbol{\mu}_{\boldsymbol{\mathcal{X}}})^t \, \boldsymbol{\Sigma}_{\mathcal{X}}^{-1} \, (\mathbf{x}_c - \boldsymbol{\mu}_{\mathcal{X}}) \leq (1.05 \, d_{\max})^2. \qquad (5)$$

Although the Gaussian prior (ROI) used here bears some similarity to [31], there are two main differences. One is that they use an SVM based classifier to process the whole face, generating several clusters each consisting of a cloud of points at the output. In contrast, we process just the most likely region obtained from the prior. Another difference is that they post-process their output using a prior to output the final location as the mean of the cluster with largest evidence. In our case, since only the features inside the prior are detected, we estimate the location of the features by using the Equation 3.

In Figure 1 (*left*), we can see the ground truth of the training samples superimposed to an image from BioID dataset. Based on the statistics of these points, we can estimate elliptical regions with high probability to contain the centers of the eyes. This region are illustrated in Figure 1 (*right*). The region of the right eye contains 814 points while the region of the left one contains 1923.

3.3. Evaluation

To assess the performance of our method, we use the accuracy measure proposed in [9]. This measure normalizes the larger displacement between the reference and the automatic label for both eyes by the correct inter-ocular distance. The error is

$$e \leq \frac{\max(e_l, e_r)}{d_{eyes}}, \qquad (6)$$

Fig. 1. *Left*:ground truth (manual annotation) of training samples superimposed to a image from BioID database. *Right*: ROI estimated from the ground truth of the training samples.

where e_l is the distance between the reference and the estimation for the left eye, e_r is this distance for the right eye and d_{eyes} is the inter-ocular distance. This error e gives us the curve labeled *worst eye*, as we show in next section.

This measure has several interesting characteristics. For $e \leq 0.25$, it is possible to locate the eye, since 25% of the inter-ocular distance is the approximate distance between the eye center and the eye corners. For $e \leq 0.10$, it is possible to locate the iris, since 10% of the inter-ocular distance is approximately the radius of the iris. With $e \leq 0.05$, it is possible to locate the pupil, since 5% of the inter-ocular distance is approximately the radius of the pupil [9, 3].

To verify the bounds on the performance of our method we also evaluate our results with two other normalized errors. The first normalized error uses the smaller error between the two eyes, while the second normalized error uses the average of the error of the two eyes. These measures give us the curves labeled *best eye* and *average eye* curves in Section 3.4.

3.4. Results in Eye Detection

In order to give performance bounds of the proposed method, we plot the hit rate curves using the three normalized errors measures (Section 3.3) in Figure 2. In Figure 3, we can see some images from BioID database with the result of the detection superimposed.

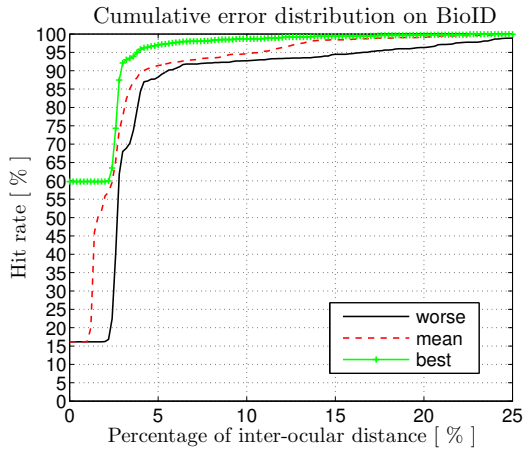

Fig. 2. Hit rate vs. normalized error for *worse eye*, *best eye* , and *average eye*, see Section 3.3.

Table 1. Performance comparison between different methods on BioID dataset. ‡ Values provided by [4]. † Values provided by [3]. § Values provided by [19]. The best and our results are in bold. (1) Using MIC+SIFT. "−" Indicate the cases in which the authors did not provide a worse eye curve and the values could not be estimated. (*) Images with closed eyes and glasses were omitted. $< \cdot >$ indicate the rank of our method in a specific accuracy.

Method	$e \leq 0.05$	$e \leq 0.1$	$e \leq 0.15$	$e \leq 0.2$	$e \leq 0.25$
Asadifard and Shanbezadeh, 2010‡ * [5]	47.0%	86.0%	89.0%	93.0%	96.0%
Asteriadis et al., 2006‡ [6]	44.0%	81.7%	92.6%	96.0%	97.4%
Bai et al., 2006† [18]	37.0%	64.00%	–	–	96.00%
Behnke, 2002‡ [17]	37.0%	86.0%	95.0%	97.5%	98.0%
Campadelli et al., 2006‡ [12]	62.0%	85.2%	87.6%	91.6%	96.1%
Chen et al., 2006‡ [14]	–	89.7%	–	–	95.7
Cristinacce et al., 2004‡ [16]	57.0%	**96.0%**	**96.5%**	97.5%	97.1%
Everingham and Zisserman, 2006§ [8]	45.87%	81.35%	–	–	91.21%
Hamouz et al., 2005‡ [15]	58.6%	75.0%	80.8%	87.6%	91.0%
Jesorsky et al., 2001‡ [9]	38.0%	78.8%	84.7%	87.2%	91.8%
Kroon et al., 2008‡ [10]	65.0%	87.0%	–	–	98.8%
Niu et al., 2006‡ [13]	75.0%	93.0%	95.8%	96.4%	97.0%
Ren et al., 2014§ [19]	77.08%	92.25%	–	–	98.99%
Timm and Barth, 2011‡ [4]	82.5%	93.4%	95.2%	96.4%	98.0%
Turkan et al., 2007‡ [11]	18.6%	73.7%	94.2%	**98.7%**	**99.6%**
Valenti and Gevers, 2012† (1) [3]	86.09%	91.67%	–	–	97.87%
IPD	**88.3%** $< 1 >$	**92.7%** $< 4 >$	**94.5%** $< 5 >$	**96.3%** $< 6 >$	**98.9%** $< 3 >$

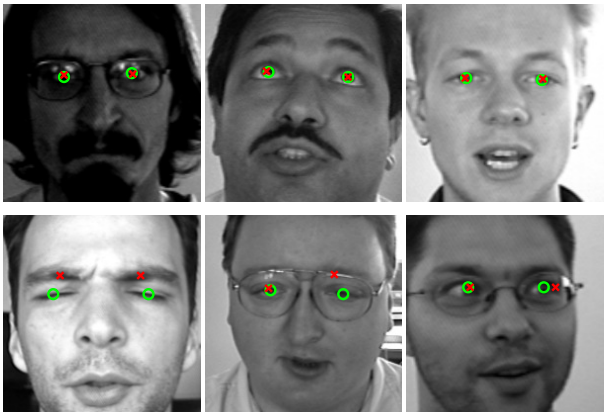

Fig. 3. Results for eye's center localization from BioID dataset. Green ball means the ground truth and red crosses are the automatic labels. Samples with good results are in the top row. In the bottom row, the bad results.

Our method is fast – we only compute inner products in low resolution images. In our experiments, the eyes were located in about 83.4 milliseconds. From this time, about 12.4 milliseconds were spent in the pre-processing, 20.5 milliseconds to locate the center of the right eye, 50.2 milliseconds to the left eye and 0.3 millisecond in the post-processing.

As we can see in Section 3.2, Figure 1, the left eye's ROI is larger than the right eye's. Then, more time was spent to locate the left eye. Another important characteristic is that the detection and the post-processing can run in parallel. In a four processors machine with hyper-threading, the cost for detecting eight points is the same of the slower one. The use of GPUs can benefit our method. We implemented our method in Matlab and ran our experiments in a machine with Intel(R) Core(TM) i7 Q950 at 3.07GHz and 8 GB DDR3/1333MHz of RAM memory.

Table 1 shows the comparison between our method and the state of the art. To train and evaluate our method, we used the BioID dataset - the standard on the literature - and the same error metric other researchers used, to compare results. Although the training/test procedures of the several papers may not have been the same, the comparison made like this is fair if we consider the methods as black boxes. We used a k-fold cross validation (results shown are the average), and the low variance show a good behavior regarding the generalization/overfitting problem. The best results and ours are shown in bold. The values highlighted by $< \cdot >$ are the ranks of our method for different values of e. For the best precision, $e \leq 0.05$, even including closed eyes and glasses we have rank 1. After that accuracy, the hit rate goes down, but remain competitive. The main advantages of our method are the good accuracy for $e \leq 0.05$ (can locate a pupil with precision) and its low computational complexity

4. CONCLUSIONS

In this paper, we describe a novel method to locate facial landmarks. The core of the proposed method is a detector based on correlation filters, that we call *Inner Product Detector* (IPD). Our technique inherits the tolerance to small variations of the desired pattern from the correlation filters and has a low computational cost. We also compare our method to the state of the art on the BioID dataset.

The IPD is linear, fast, and achieves very competitive results. A very important factor is that, different than a lot of the state-of-the-art algorithms, there is no global model to improve the location of the detected points using the information of the rest of the face.

As future work, we plan to have multiple trainings working together to address the multiple appearance scenario.

5. REFERENCES

[1] Andrew T. Duchowski, *Eye Tracking Methodology: Theory and Practice*, Springer, 2nd edition, July 2007.

[2] R. Bates, H. Istance, L. Oosthuizen, and P. Majaranta, "Survey of de-facto standards in eye tracking.," in *COGAIN*, 2005.

[3] R. Valenti and T. Gevers, "Accurate eye center location through invariant isocentric patterns," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 34, no. 9, pp. 1785–1798, Sept 2012.

[4] Fabian Timm and Erhardt Barth, "Accurate eye centre localisation by means of gradients," in *VISAPP*, 2011.

[5] Mansour Asadifard and Jamshid Shanbezadeh, "Automatic adaptive center of pupil detection using face detection and cdf analysis," 2010, IMECS.

[6] S.Asteriadis, N.Nikolaidis, A.Hajdu, and I.Pitas, "An eye detection algorithm using pixel to edge information," 2006, EUSIPCO.

[7] Zhi-Hua Zhou and Xin Geng, "Projection functions for eye detection," *Pattern Recognition*, vol. 37, no. 5, pp. 1049–1056, May 2004.

[8] M. Everingham and A. Zisserman, "Regression and classification approaches to eye localization in face images," in *Automatic Face and Gesture Recognition, 2006. FGR 2006. 7th International Conference on*, April 2006, pp. 441–446.

[9] Oliver Jesorsky, Klaus J. Kirchberg, and Robert Frischholz, "Robust face detection using the hausdorff distance," in *AVBPA*, 2001.

[10] Bart Kroon, Alan Hanjalic, and Sander M.P. Maas, "Eye localization for face matching: is it always useful and under what conditions?," in *CIVR*, 2008.

[11] Mehmet Türkan, Montse Pardàs, and A. Enis Çetin, "Human eye localization using edge projections.," in *VISAPP*, 2007.

[12] Paola Campadelli, Raffaella Lanzarotti, and Giuseppe Lipori, "Precise eye localization through a general-to-specific model definition," in *BMVC*, 2006, pp. 187–196.

[13] Zhiheng Niu, Shiguang Shan, Shengye Yan, Xilin Chen, and Wen Gao, "2d cascaded adaboost for eye localization," in *Pattern Recognition, 2006. ICPR 2006. 18th International Conference on*, 2006, vol. 2, pp. 1216–1219.

[14] Dan Chen, Xusheng Tang, Zongying Ou, and Ning Xi, "A hierarchical floatboost and mlp classifier for mobile phone embedded eye location system," in *ISNN (2)*, 2006, pp. 20–25.

[15] M. Hamouz, J. Kittler, J.K. Kamarainen, P. Paalanen, H. Kalviainen, and J. Matas, "Feature-based affine-invariant localization of faces," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 27, no. 9, pp. 1490 –1495, sept. 2005.

[16] D. Cristinacce, T. Cootes, and I. Scott, "A multi-stage approach to facial feature detection," in *BMVC*, 2004.

[17] Sven Behnke, "Learning face localization using hierarchical recurrent networks," in *ICANN*, 2002.

[18] Li Bai, Linlin Shen, and Yan Wang, "A novel eye location algorithm based on radial symmetry transform," in *ICPR*, 2006.

[19] Yan Ren, Shuang Wang, Biao Hou, and Jingjing Ma, "A novel eye localization method with rotation invariance," *Image Processing, IEEE Transactions on*, vol. 23, no. 1, pp. 226–239, Jan 2014.

[20] P.N. Belhumeur, D.W. Jacobs, D.J. Kriegman, and N.; Kumar, "Localizing parts of faces using a consensus of exemplars," in *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, 2011, pp. 545–552.

[21] B. V. K. Vijaya Kumar, Abhijit Mahalanobis, and Richard D. Juday, *Correlation Pattern Recognition*, Cambridge University Press, New York, NY, USA, 2005.

[22] David S. Bolme, J. Ross Beveridge, Bruce A. Draper, and Yui Man Lui, "Visual object tracking using adaptive correlation filters," in *CVPR*, 2010, pp. 2544–2550.

[23] D.S. Bolme, B.A. Draper, and J.R. Beveridge, "Average of synthetic exact filters," in *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, June 2009, pp. 2105–2112.

[24] Chunyan Xie, Marios Savvides, and B.V.K. Vijaya Kumar, "Redundant class-dependence feature analysis based on correlation filters using frgc2.0 data," *Computer Vision and Pattern Recognition Workshop*, vol. 0, pp. 153, 2005.

[25] Hung Lai, Venkatesh Ramanathan, and Harry Wechsler, "Reliable face recognition using adaptive and robust correlation filters," *Computer Vision and Image Understanding*, vol. 111, no. 3, pp. 329 – 350, 2008.

[26] Gabriel Araujo, Waldir Sabino, Eduardo da Silva, and Siome Goldenstein, "Facial landmarks detection based on correlation filters," in *International Telecommunications Symposium (ITS)*, 2010.

[27] Gabriel M. Araujo, Eduardo A. B. Silva, Alexandre G. Ciancio, José F. L. de Oliveira, Felipe M. L. Ribeiro, and Amir Said, "Integration of eye detection and tracking in videoconference sequences using temporal consistency and geometrical constraints," in *IEEE International Conference on Image Processing (ICIP)*, 2012.

[28] BioID Technology Research, "The bioid face database," http://www.bioid.com, 2011.

[29] Paul A. Viola and Michael J. Jones, "Robust real-time face detection," *International Journal of Computer Vision*, vol. 57, no. 2, pp. 137–154, 2004.

[30] Xiaoyang Tan and Bill Triggs, "Enhanced local texture feature sets for face recognition under difficult lighting conditions," in *AMFG*, 2007.

[31] Ognjen Arandjelovic and Andrew Zisserman, "Automatic face recognition for film character retrieval in feature-length films," in *CVPR*, 2005.