Precise multi-level face detector for advanced analysis of facial images

Article *in* IET Image Processing · March 2012

DOI: 10.1049/iet-ipr.2010.0495

Michal Kawulok
Silesian University of Technology

Precise multi-level face detector for advanced analysis of facial images

Michal Kawulok[1] and Janusz Szymanek[2]

[1]Institute of Informatics, Silesian University of Technology,
Akademicka 16, 44-100 Gliwice, Poland

[2]Future Processing, Research Lab, Rokitnicka 67, 41-936 Bytom,
Poland

July 12, 2011

Abstract

This paper presents a new method for precise detection of frontal human faces and eyes using a multi-level ellipse detector combined with a support vector machines verifier. Our main contribution lies in improving the accuracy of eye detection in high-quality images, which is often neglected by alternative methods. Although many approaches to face detection have been proposed recently, relatively little attention has been paid to the detection precision. It is worth noting that the detection precision is particularly important for face analysis purposes. More specifically, we demonstrate that the detection error propagation substantially affects the face recognition performance. With the proposed improvements we have managed to increase the face recognition rate by 7.7% for AR database compared with the publicly-available implementation of the well-established Viola-Jones face and eye detector.

1 Introduction

Face detection is a commonly performed task in our daily communication and it is one of the most investigated areas of computer vision [1]. The face detection aim is to determine whether there are any faces in a considered image and in the case of a positive answer, to find their exact locations.

Face detection has found a broad range of applications including the surveillance tracking, human-computer interfaces and entertainment purposes. It is also the first, but a very important step of automatic face identification [2, 3], gender classification [4] and other forms of face analysis. It is worth noting that many applications require only a rough estimate of where the faces are located. In general, these are the cases when the detection result is supposed to be further analysed by human observers or when the task is to count the faces in the image. This explains why the majority of algorithms are evaluated using images or videos with low quality, in which the faces are hardly visible, for example the MIT+CMU data set [5].

However, if the detection result serves as an input for different face image processing algorithms (e.g. face recognition), high detection precision is essential for proper normalisation of the images. If the facial features are imprecisely located, then after normalisation the faces may be tilted or misplaced, which usually makes further analysis ineffective. The presented experimental validation demonstrates that the face detection precision has a crucial influence on the effectiveness of face recognition. The vast majority of works on face recognition are not focused on detecting the faces, and assume that the face location is known. On the other hand, relatively little attention has been paid to the detection precision among the many works on face detection. If the faces are oriented frontally, which is assumed in the work reported here, their position can be defined unambiguously by central points of eyes. We propose a new method for improving the face detection focused on maximising the eye detection precision, which is particularly relevant for face analysis purposes.

We detect the faces in the luminance channel using geometric properties of the human faces observed at different levels. We utilised and developed the idea proposed by D. Maio and D. Maltoni for face localisation [6] using the elliptical features. We have extended their method to detect not only the human faces, but also the eye sockets and irises. Moreover, we use the support vector machines (SVM) verifier to eliminate false positives at every level of the analysis.

The paper is organised as follows. Alternative approaches to face detection are briefly described in Section 2. Section 3 presents the ellipse detection algorithm used in Maio and Maltoni's work [6]. The detailed description of our method is given in Section 4, and the experimental results are shown and discussed in Section 5. Section 6 concludes the paper.

2 Related work

There are many different techniques for face detection and localisation which utilise common properties of human faces [1]. A number of methods use heuristic rules to define a face or facial features. These rules can subsequently be applied in a top-down manner to narrow down the searched area and potential locations of faces [6, 7]. Alternatively, it is possible to detect facial features first (e.g. the eyes using morphological operations [8]) and then intend to match them into potential face regions.

An example of a top-down method is the one presented by D. Maio and D. Maltoni [6]. In this approach, first the ellipses are detected using the generalised Hough transform [9] to find approximate face locations. After that, the vertical direction profile is computed within each ellipse to determine the position of eyes and mouth (which are characterised by many horizontally-oriented edges), and verify whether the ellipse represents a human face. While the first step is very successful in finding the face candidates, the latter performs well only for vertically-oriented faces, in which two local maxima in the direction profile indicate the positions of mouth and eyes. However, this method is usually ineffective if a face is even slightly tilted or if its expression is not neutral.

There are also many algorithms which detect faces and skin regions based on skin color, using either parametric models which implement fixed decision rules [10] or statistical models which require appropriate training [11].

Face appearance can also be learned from a representative training set to make it possible to determine whether a given image region contains a face [5,12].

This is often effective, but the whole image must be analysed with a varying size of the scanning window, which seriously affects the performance and limits real-time applications. P. Viola and M.J. Jones have overcome this disadvantage by applying a cascade of fast simple classifiers based on Haar-like features optimised using AdaBoost technique [13]. It is worth noting that this detector gained popularity in face detection field, but it can be used for detecting any kind of objects. This method was further improved [14,15] to decrease required number of classifiers in the cascade, and it is still regarded as the leading face detector. This detector was implemented in a popular Open Computer Vision library (OpenCV) [16].

There are also many works focused on rotation invariant detectors and multi-view representations of the human faces [17, 18]. In majority, these are extensions of the Viola-Jones detector, using cascades of classifiers for generalised cases. There are also attempts to utilise colour-based detectors for fast preliminary selection, and then combine them with the Viola-Jones detector [19] or template-based methods [20]. Recently, W.-K. Tsao et al. [21] presented a face detector which works by finding feature patterns using data mining techniques.

There are also many works focused on detecting facial features like eyes, mouth, and nose inside a given face region. Eye detection techniques are of a particular concern here as they may contribute to increasing the face detection precision. D.W. Hansen in his recent survey on eye modelling [22] divided eye detection techniques into three main categories, namely: (1) shape-based, (2) feature-based, and (3) appearance-based methods. Shape-based techniques utilise circular and elliptical shapes and curvatures in the eye region to locate the eye-lids, iris and pupil [23, 24]. These methods are usually quite precise, but they fail to detect eyes which are not wide open. Feature-based methods analyse local image features that are specific to the eye region. They are usually extracted using morphology-based techniques [10, 25] or filter responses [26]. Appearance-based methods operate using an eye template defined either in the intensity domain [27] or using subspace methods [28]. Among the appearance-based methods, Viola-Jones object detector is also successfully used for eye detection [29, 30]. These detectors are heavily dependent on training, during which a cascade of simple Haar features-based classifiers is generated [31].

Within the most important problems concerned with the existing eye detectors are those related to sensitiveness to the acquisition conditions (lighting and face pose), and also scale inflexibility. Many detectors perform very well at a certain scale, but they are not applicable if the eye region is too small or when the algorithm is started in an unexpected position. In this context, the proposed face detection system allows for detecting face and facial features at different scales, using shape-based and appearance-based algorithms.

3 Ellipse detection

This Section presents the ellipse detector [6] which we used in our algorithm. In general, the ellipses are detected by performing the Hough transform [9] in a directional image. In order to obtain the directional image, averaged tan-

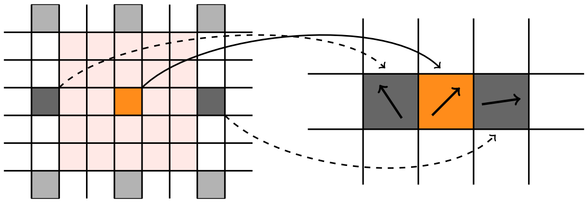


Figure 1: *Procedure for determining tangent vectors. For every ninth pixel in the original image (left) a single tangent vector is obtained (right) based on* 5×5 *neighbourhood.*

gent directions for every 3×3 group of gradient vectors are approximated. The gradient vector $\mathbf{g}_{m,n}$ is computed for every pixel $I_{m,n}$ using Sobel operator:

$$\mathbf{g}_{m,n} = \begin{bmatrix} I_{m+1,n+1} + I_{m+1,n-1} - I_{m-1,n+1} - I_{m-1,n-1} + 2I_{m+1,n} - 2I_{m-1,n} \\ I_{m+1,n-1} + I_{m-1,n-1} - I_{m-1,n+1} - I_{m+1,n+1} + 2I_{m,n-1} - 2I_{m,n+1} \end{bmatrix}. \quad (1)$$

After that, for every third pixel in every third row (i.e. for every ninth pixel in the image) a single tangent vector ($\mathbf{u}_{m,n}$) is determined based on 25 gradients in 5×5 neighbourhood. This is illustrated in Fig. 1. For the orange pixel in the original image (left) a tangent direction is obtained (right) based on the pixel's neighbourhood (light orange). The procedure is repeated for every third pixel in every third row (grey shade).

The tangent vector is a unit vector which minimises the given error function:

$$\delta(\mathbf{u}_{m,n}) = \sum_{i=m-2}^{m+2} \sum_{j=n-2}^{n+2} (\mathbf{g}_{i,j} \cdot \mathbf{u}_{m,n})^2. \quad (2)$$

The function is minimised analytically to obtain the tangent angle. If variance of the gradient directions is high within the group of pixels, the normalised error of averaging defined as:

$$E_{m,n} = \delta_{min}(\mathbf{u}_{m,n}) \Big/ \sum_{i=m-2}^{m+2} \sum_{j=n-2}^{n+2} (\mathbf{g}_{i,j})^2 \tag{3}$$

is high as well and the detected tangent direction is not reliable. The directions, for which the error is greater than a certain threshold value (E_{th}), are considered unreliable and are rejected. Every detected tangent direction is characterised by its location, angle, and magnitude defined as a sum of squared contributing gradients. The magnitude quantifies how strong the direction is, and only the directions with high magnitudes are considered further.

The generalised Hough transform [9] is a technique for detecting parameterised shapes in digital images. It works by transforming the input image (usually a directional image) to a parameter space (*accumulator*). Every pixel or direction is considered as a part of a potential shape and contributes to the accumulator by adding the most probable values of the parameters. Local maxima in the accumulator indicate the most probable values of the parameters which define the shape location.

In the analysed case ellipses of a fixed size (defined by the length of the semi-axes a and b) are detected and their locations in the image (i.e. the ellipses' central points) are regarded as parameters. Hence, the parameter space is two-dimensional, and the accumulator shows the most probable locations of the ellipse centres. Although the ellipse size is fixed, it is treated with tolerance. Lengths of the semi-axes are defined as ranges rather than fixed values ($a_{min} = \rho_r a$, $a_{max} = \rho_e a$, $b_{min} = \rho_r b$, $b_{max} = \rho_e b$, where ρ_r and ρ_e are reduction and expansion factors). This is illustrated in Fig. 2(a).

It is assumed that every tangent direction can belong to two potential ellipses, so each direction should contribute to two exclusive areas (which correspond to ranges of probable ellipse centres) in the accumulator. For computation reasons, a set of predefined templates is prepared for 256 discrete directions before starting the processing. Examples of the templates generated for different angles are presented in Fig. 2(b). Hence, for every direction an appropriate template is picked from the set and added to the accumulator at the direction's location. Every template contains values in range $[0, 1]$, and values in the accumulator are usually below 100. When the accumulator is generated, ellipse central points are determined as local maxima of the accumulator whose values exceed a given threshold value (we used 50 for ellipse detection). If it is necessary to detect ellipses of different absolute sizes, the image is scaled in a pyramid manner, so as to use always the same set of templates.

4 Proposed algorithm for face and eye detection

The proposed face detection algorithm is applied at three main levels: 1) whole image level: detection of potential face locations (called *face candidates*), 2) head-ellipse level: verification of the candidates, and 3) eye socket level: precise detection of eye centres. Outline of the algorithm is presented in Fig. 3. At the first level, we perform ellipse detection as described in Section 3 to find potential areas, where human faces may be located. Then, at the second level, the eye sockets are found within the detected head-ellipses. This is achieved using a

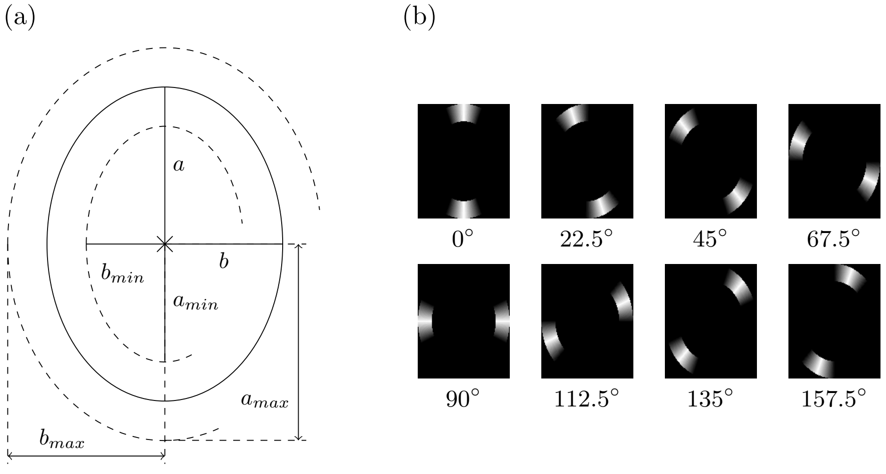
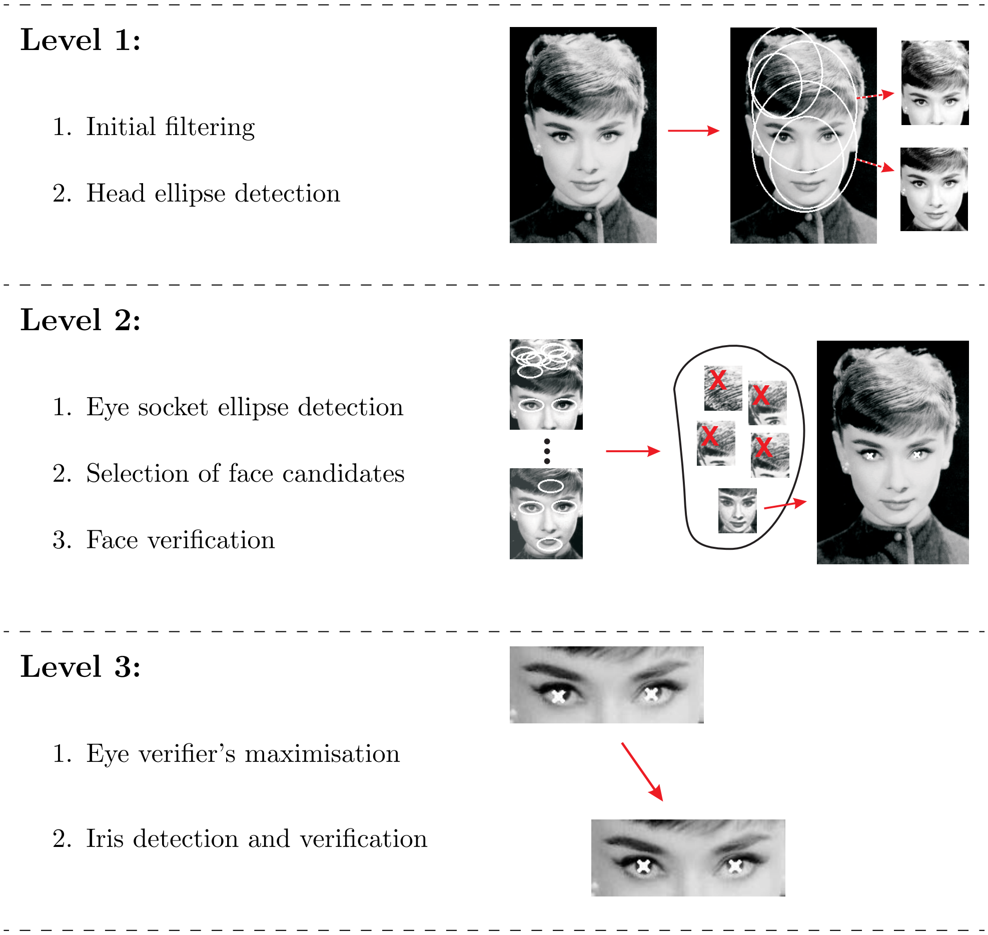


Figure 2: *Ellipse representation for the Hough transform (a) and templates generated for directions of different angles (b).*

modified ellipse detector described later in this Section. The potential eyes are paired to generate face candidates, which are verified using an SVM classifier. Finally, at the third level, eye detection techniques are applied. Appearance-based procedure for iterative eye position correction and verification is used to find a local maximum of the SVM eye verifier's response. Furthermore, the shape-based iris detection is performed for precision improvement.

4.1 Detection of head ellipses

For head ellipse detection we adopted approach described in Section 3 and the whole process is illustrated in Fig. 4. First, the original image (a) is processed with 3×3 median and Gaussian filters in order to minimise false edge detection



Level 1:

1. Initial filtering
2. Head ellipse detection

Level 2:

1. Eye socket ellipse detection
2. Selection of face candidates
3. Face verification

Level 3:

1. Eye verifier's maximisation
2. Iris detection and verification

Figure 3: *Proposed multi-level algorithm for face detection.*

due to noise (b). Otherwise, noisy pixels would introduce strong gradients which results in finding false tangent vectors. Later, such vectors would contribute to the accumulator and affect the ellipse detection. Although there are many more advanced techniques for noise reduction, we found this simple operation sufficient to substantially reduce false ellipse detection.

Figure 4(c) presents a thresholded magnitude image of determined tangent vectors (with $E_{th} = 0.25$), and (d) shows the corresponding tangent directions.

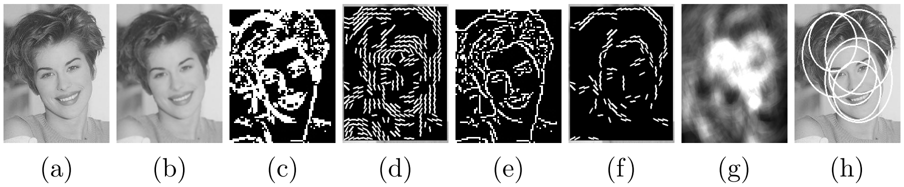


Figure 4: *Selected steps of head ellipse detection: original image (a), filtered image (b), directional image before (c, d) and after (e, f) Canny edge detector-based elimination, accumulator (g) and detected ellipses (h).*

It can be observed from the figure that there are many parallel vectors which are not beneficial for detecting ellipses. Therefore, the magnitude image (c) is filtered using Canny edge detector to eliminate those redundant directions. The magnitude image after filtering is presented in (e) and corresponding tangent directions are shown in (f). Only those positively verified tangent vectors contribute to the accumulator (g). Here, we search for ellipses of size 32×40 pixels with reduction and expansion coefficients $\rho_r = 0.8$ and $\rho_e = 1.15$. Depending on minimal and maximal relative size of faces that are to be detected, the image is scaled in a pyramid manner from the smallest to the largest, so that all sizes of ellipses are covered. In most cases related to automatic face analysis it is sufficient to search for faces of size from 10% to 100% of the image larger dimension.

4.2 Face candidates extraction and verification

A rectangular bounding box of every detected ellipse is clipped and scaled to a size of 120×150 pixels, and histogram of the resulting image is equalised.

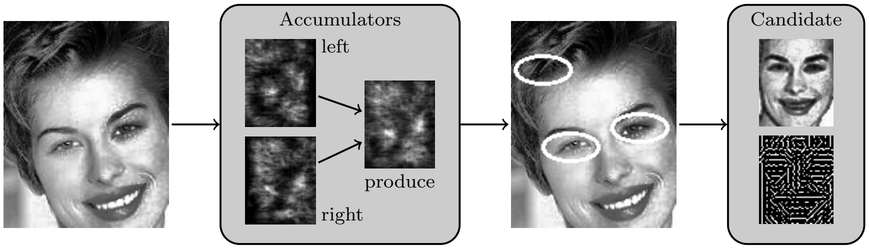


Figure 5: *Eye socket detection using Hough transform and generation of a face candidate based on locations of detected ellipses.*

Inside these rectangles the ellipse detection is repeated to find the candidates for eye sockets. Eye sockets are characterised by elliptical shapes, but it may be observed that the edges are stronger at the inner face sides, near the nose. Based on this observation, the ellipse detection procedure used for head detection has been modified. First, half-ellipses (right and left) are detected in the image and two accumulators (left and right) are created separately, which is shown in Fig. 5. The detection is performed for half-ellipses of a constant size of 10×5 pixels with $\rho_r = 0.3$ and $\rho_e = 1.3$. After that, the central points of ellipses are extracted as local maxima in three accumulators: left, right and in the produce of the two. The produce is used to improve detection of whole ellipses (it works as a full-ellipse accumulator). Although our experiments indicated that using half-ellipses is better, some eye-socket ellipses can be detected only from this full-ellipse accumulator.

Central point of every detected eye-socket ellipse is considered as an eye candidate. After that, every pair of eye candidates is regarded as a potential face and it is verified against simple heuristic rules: 1) distance between the points must be greater than 25% of the width of the head ellipse's bounding box (i.e. the length of the minor axis), and 2) an angle of the line which crosses these points must be smaller than 30°. Hence, for each head ellipse a set of face candidates (pairs of eye centres) is prepared.

Every face candidate is subjected to a double verification using SVMs [32]. Here, SVM is first trained with two classes of normalised images representing the positive and negative cases. For training and classification we used radial basis function (RBF) kernel. Its value is computed for two vectors $\mathbf{v}_1$ and $\mathbf{v}_2$ as

$$K(\mathbf{v}_1, \mathbf{v}_2) = \exp\left[-\frac{\|\mathbf{v}_1 - \mathbf{v}_2\|^2}{\sigma^2}\right], \tag{4}$$

where σ is a width parameter. In order to accept the candidate, the result of the following two verification steps must be positive:

1. Verification of the intensity images. The image normalisation includes affine transforms, after which the central points of eyes are located in fixed positions, and histogram equalisation which is a commonly adopted technique in face processing [2]. At the end, the images are downsampled to 16×19 pixels size. Subsequent rows are stacked to form a single vector which is processed by SVM (here we used RBF kernel with $\sigma = 0.9$). Examples of normalised face candidates used for training are presented in Fig. 6 before (a, c) and after (b, d) the downsampling. Positive cases (a, b) were normalised based on real eye positions determined by human observers and false candidates (c, d) were obtained as a result of ellipse detection in complex background.

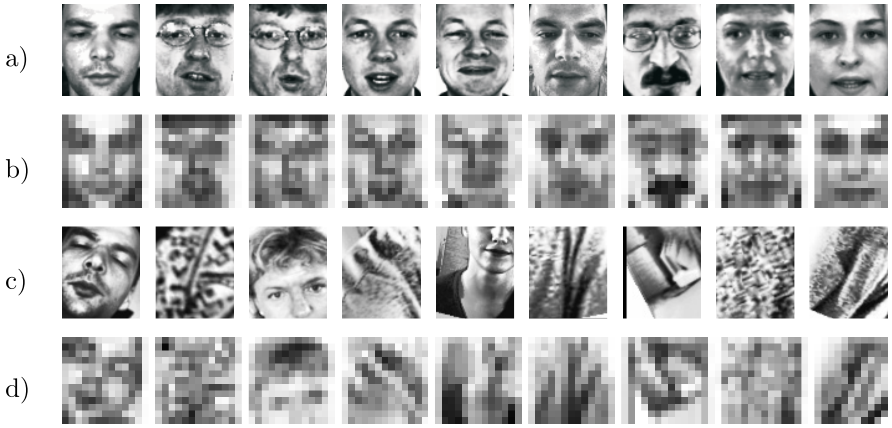


Figure 6: *Examples of images used for SVM training: positive face candidates before (a) and after (b) downsampling, and non-faces before (c) and after (d) downsampling.*

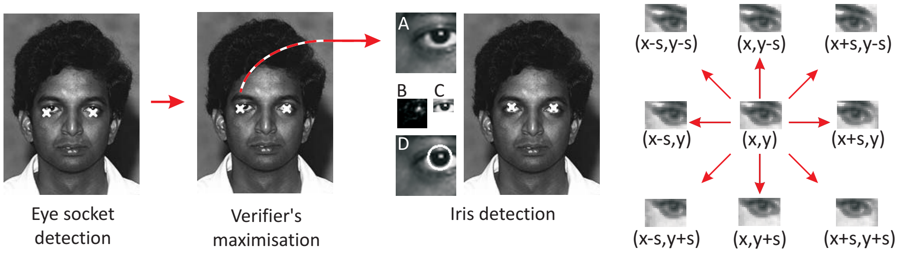
2. Verification of a directional image. Here we use the geometrically normalised images of size 64×75 as shown in Fig. 6(a, c), which are converted to a directional image of size 20×24, following the procedure described in Section 3. An example of the candidate's intensity and directional image is presented in Fig. 5 (right). Every tangent vector (characterised by direction and magnitude) is converted to the Cartesian coordinate system ($\mathbf{u} = [x, y]$). All of these vectors ($\mathbf{u}_1, ..., \mathbf{u}_n$, where $n = 480$) are stacked to form a vector ($\mathbf{v} = [x_1, y_1, ..., x_n, y_n]$) which is subject to the second verification (RBF kernel with $\sigma = 3$ was used).

4.3 Precise eye centres detection

If a face candidate is accepted, the eye positions are determined as centres of the eye-socket ellipses detected at the second level. This is a good approximation, however it is insufficient if the eyes are expected to be localised precisely. Because of this, at the final level we perform two operations aimed at detecting eye centres: a) eye verification maximisation, and b) iris detection followed by iris verification.

The first operation is an iterative appearance-based procedure aimed at finding a local maximum of the eye verifier. For the verification SVM with RBF kernel ($\sigma = 1$) is used. It is trained with eye and non-eye images normalised to a size of 16×24 pixels. At every i-th step of the procedure the current eye position p_i is modified in eight directions by a step s_i as presented in Fig. 7(b). Eye image is cropped at every modified position and verified. If the verifier's response for a position p_n is greater than the response in the current position p_i, it is moved to the new position ($p_{i+1} = p_n$). If the current position gives the maximal response, the step is decreased ($s_{i+1} = s_i/2$) and the position remains unchanged ($p_{i+1} = p_i$). The process is started with the step $s_0 = 4$ and repeated until it equals 1. As a result, the eye position is moved to a local maximum of the verifier's response. This position is closer to the real location than a central point of the eye-socket ellipse.

The second operation of the precision improvement is aimed at locating the irises, which is illustrated in Fig. 7(a) (Iris detection). This is performed by detecting circles in the images of eyes normalised to a size of 72×72 pixels (A in



(a) (b)

Figure 7: *Procedure for improving eye precision (a) and eye position modification using verifier's maximisation (b).*

the Figure), which we found sufficient for the detection procedure. Here, a square area around the eye position of size equal to 0.45 of the distance between the eyes is processed. This region contains the iris even if the eye was detected with low precision, but it should not present any other circular objects. The iris detection is implemented again using the ellipse detector with parameters: $a = b = 7$, $\rho_r = 0.8$, and $\rho_e = 1.15$. These values correspond to the expected size of an iris in the normalised image (A). The eye position is moved to the maximal value of the accumulator (B) and the iris image (C) is verified with SVM. For a positive response, the eye position is updated to the centre of the circle (D). Otherwise, it is assumed that the detected circle is not an iris (which may be not visible or may be occluded by glasses). Even a partially-visible irises can be detected using this technique, and it may be verified positively assuming that the verifier was trained from images capturing various eye poses.

5 Experimental validation

Usually face detection effectiveness is measured with detection and false positive rates. The first one (η_{face}) is a percentage of faces that were correctly detected in the analysed set of images, while the latter (δ_{FP}) is a percentage of non-faces within the detected faces. This is an appropriate metric as long as the aim is to count the faces in images or give a rough approximation of face location. However, for face analysis purposes not only does it matter whether the face was found, but also how precisely it was located. If the detection precision error may be propagated further, the detector's evaluation should take the precision into account. Therefore, for every correctly detected face we define the relative precision error similarly to O. Jesorsky [33] as:

$$\delta_d = \frac{\Delta_l + \Delta_r}{2D}, \tag{5}$$

where Δ_l and Δ_r are, respectively, the absolute distances in pixels between the real and detected positions for left and right eye, and D is the distance between the real positions of eyes.

We implemented our algorithms in C++ using Adaptive Vision Studio environment (available at http://www.adaptive-vision.com) and the tests were performed on 2GHz Pentium processor. During the validation we compared our approach with the Viola-Jones face and eyes detectors improved by R. Lienhart [14], and implemented in the OpenCV library [16]. We used the publicly available trained classifier for face detection (*haarcascade_frontalface_alt2.xml*) with two alternative eye classifiers: (1) *haarcascade_eye.xml* (included in the

OpenCV release), and (2) *ojoD.xml* for right and *ojoI.xml* for the left eye prepared by M. Castrillón-Santana [29], which are included in ENCARA2 face detection system.

We used face images from FRGC Version 1 database [3] for training purposes and validated our algorithm using 3657 frontal face images from Feret database [34], 3313 images from AR database [35] and MIT+CMU test set [5] (file lists are available at `http://sun.aei.polsl.pl/~mkawulok/face_detection/file_lists.txt`). We investigated detection effectiveness given by false positives and false negatives, the detection precision (5) and time of processing. Additionally, for Feret and AR we measured dependence between face detection precision and face recognition rate.

5.1 Parameters and processing time

The algorithm depends on many parameters which were determined experimentally. Here we focus on two most important ones concerned with the head ellipse detection: a) minimal size of heads (relative to the image height) and b) ellipse template size. The head ellipse detection is performed at the very beginning, and has a large influence on the execution time. The aforementioned parameters determine maximal image size to which the input image must be scaled and number of steps in the pyramid ellipse detection as pointed in Section 4.1.

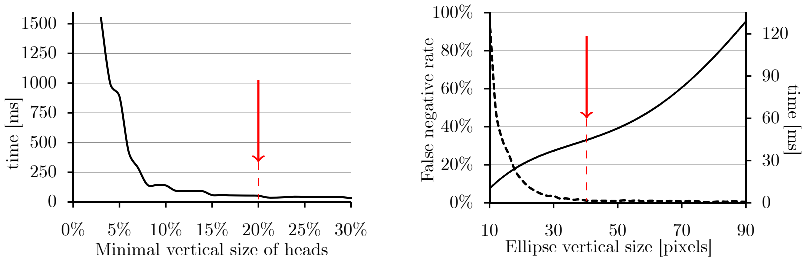
The dependence between the minimal vertical head size and the execution time is shown in Fig. 8(a). The processing time is small for faces whose height is greater than 10% of the image height. However, if small faces (smaller than

10% of the image height) are to be detected, the necessary processing time increases drastically. Interestingly, size of the input images has little influence on the performance, because the images are scaled to a fixed size depending on the ellipse template size. Also, the processing time does not depend on number of faces which are in the image as the most time-consuming task is the generation of a directional image and the head ellipse detection. Number of detected candidates and faces affect number of verifications, but it has negligible influence on the overall performance.

The ellipse template size influences both the processing time and the false negative rate. This dependence (obtained for the minimal vertical size set to 20%) is illustrated in Fig. 8(b). With decreasing template size fewer faces can be detected, which results in growing number of false negatives. On the other hand, when larger templates are used, the input image must be scaled to a larger size to detect small ellipses. For ellipse size of 32×40 (indicated with an arrow in the graph) false negatives stabilise at a low value, while the time is still acceptable (52 ms). This means that the first image in the pyramid is scaled to a height of 200 pixels. Such configuration allows the detector to work at 20 frames per second which is suitable for real-time applications. This is a similar result to that achieved using the Viola-Jones detector.

5.2 Detection effectiveness and precision

All results presented in this Section were obtained with the same values of parameters determined experimentally as presented in Section 5.1. The only



(a) (b)

Figure 8: *Influence of ellipse detection parameters (minimal size of heads (a) and ellipse template size (b)) on time and effectiveness.*

exception was done for tests on MIT+CMU database. This set contains many faces which are smaller than 10% of the image height, so the minimal size of head ellipses was decreased as well.

The results of face detection effectiveness and precision achieved using our method and two detectors implementing Viola-Jones cascades are presented in Tab. 1 (OCV for OpenCV detector [16] and E2 for ENCARA2 eye classifier [29]). OCV and E2 differ only in the eye classifier used, which explains identical face detection scores (η_{face} and δ_{FP}). The best results are highlighted for every case, and it can be seen that our detector performs much better in terms of the eye detection rate (η_{eyes}) and precision error (δ_d) for all of the tested data sets. The advantage over the Viola-Jones detectors in terms of the eye detection rate (η_{eyes}) is 10.9% for Feret and 22% for AR (third row). It can be observed from the Table that our method gives worse face detection scores than the Viola-Jones detectors for MIT+CMU data set. As it was mentioned, this set contains many small faces of poor quality. Eye sockets are hardly visible in these small, low-quality images, which is the main reason for the observed drop in the face detection rate. However, it is worth noting that even for MIT+CMU data set our detector renders much higher eye detection rate and detection precision.

Table 1: *Comparison of face and eye detection rates (OCV - OpenCV face and eye detector [16], E2 - ENCARA2 eye classifier [29]).*

	MIT+CMU			Feret			AR		
	Our	OCV	E2	Our	OCV	E2	Our	OCV	E2
η_{face}	60.4%	**79.2%**	**79.2%**	99.6%	**99.8%**	**99.8%**	**89.2%**	88.5%	88.5%
δ_{FP}	24.5%	**5.7%**	**5.7%**	**0.1%**	1.2%	1.2%	**0.3%**	4.4%	4.4%
η_{eyes}	**60.4%**	16.0%	11.2%	**99.6%**	88.7%	88.3%	**89.2%**	67.2%	57.13%
δ_d	**10.89%**	18.86%	21.13%	**4.04%**	5.62%	5.2%	**5.16%**	6.59%	8.66%

It can be concluded that in case of high quality images our detector is competitive compared with other state-of-the-art methods, especially in terms of the detection precision. Later, in Section 5.3 we show that this difference is substantial, if the detection results are to be used for further face analysis. These experiments identified the weak points of our method which helped us define the minimal requirements. The method is applicable only to the images, in which the face size is greater than 10% of the image height, and ideally the faces should cover area larger than 120×150 pixels. It is worth noting that this condition is met by the majority of available databases designed for face analysis evaluation. If it is required to process images with smaller faces, it would be worth considering to use an alternative face detector instead of two first levels of our method to find approximate location of faces. After that, the third level could be applied to improve the detection precision. Low precision error obtained by our method for MIT+CMU data set confirms sensibility of such approach.

Four examples of detected eyes and faces are presented in Fig. 9. For every pair of images the left one presents the result obtained using the OpenCV detector [16], and the right one – using our detector. Although the faces were correctly found for all the cases with both methods, it can be observed that the eye detection precision is noticeably higher for our detector. In (b) and (d) the irises were not detected using our method as they are not visible, so the eye positions were obtained from the eye-socket centres improved with the verification maximisation. In the remaining images the irises were found and positively verified. In some cases, as in (d), the OpenCV face detector succeeds, but the eye classifier fails to detect the eyes.

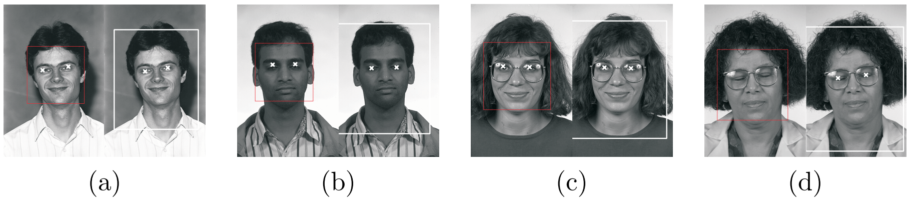

(a) (b) (c) (d)

Figure 9: *Detection results obtained using the OpenCV cascades (left in each pair) and our detector (right in each pair).*

5.3 Detection precision and face recognition rate

A very important part of the experimental validation was focused on investigating the dependence between the detection precision and the face recognition score. We used the Fisherfaces method [36] for face recognition, trained with the images from FRGC Version 1 database [3]. The face recognition accuracy was evaluated for images from Feret and AR databases. Every data set was randomly split into a gallery which contains one image per person and a query set which contains images of people whose images are in the gallery. For every image in the query set the most similar image from the gallery was found to create the first-match pair. The classification was regarded as correct if this pair of images presented the same individual. The effectiveness of face recognition (η_{rec}) is defined as a percentage of correctly classified face images from the query set.

The detection precision (δ_d) for different cases and the corresponding face recognition effectiveness (η_{rec}) are presented in Tab. 2. The experiments were conducted for the following cases: a) eye-socket detection obtained at the second level of our algorithm without any precision improvement (ED), b) after the verifier's maximisation correction (VC) and c) with the iris detection (ID). Additionally, the recognition rate was measured for real eye positions determined by human observers (RE) and compared with the results obtained using the OpenCV (OCV) and ENCARA2 (E2) detectors. AR database contains many difficult cases (occlusions, varying face expression, sunglasses) which explains worse detection and recognition rates compared with Feret. Our method outperforms the alternative detectors in terms of detection precision and this allows to increase the recognition scores by 0.5% for Feret and 7.7% for AR.

The detection precision has a significant influence on the face recognition score, which is presented in Fig. 10. This strong dependence is explained by

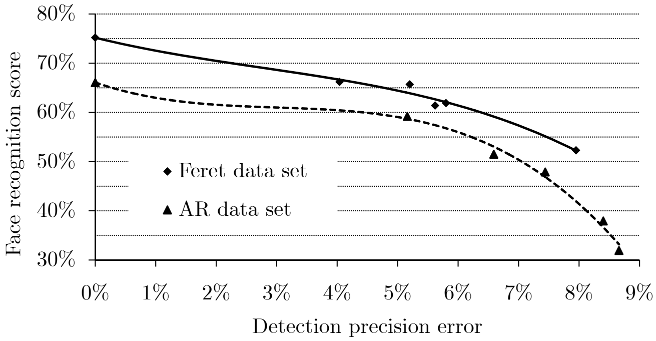


Figure 10: *Relation between the detection precision and face recognition score.*

the fact that face recognition algorithms rely on geometrically normalised face images, in which the eyes are in fixed positions. Hence, even for slight eye position displacement, different features are extracted, which affects the face recognition effectiveness.

Table 2: *Detection precision and its influence on face recognition for Feret and AR data sets.*

		OCV	E2	ED	VC	ID	RE
Feret	δ_d	5.62%	5.2%	7.95%	5.8%	**4.04%**	-
	η_{rec}	61.4%	65.7%	52.3%	61.9%	**66.2%**	75.2%
AR	δ_d	6.59%	8.66%	8.4%	7.44%	**5.16%**	-
	η_{rec}	51.5%	31.07%	37.94 %	47.89%	**59.2 %**	66.07%

Our algorithm was designed and optimised for the face analysis purposes, which means that it detects the human faces and eyes with high precision in images of quality sufficient for further processing. Images of poor quality usually cannot be recognised correctly, so this drawback of the proposed detector does not downgrade the overall effectiveness of a potential face recognition system.

6 Discussion and conclusions

In this paper we introduced a new multi-level algorithm for fast and precise face and eye detection using elliptical features. The eye-socket detection combined with the precision improvement techniques are the main contribution of the work reported here. Our method obtains much better results in terms of the detection precision compared to alternative leading face detectors, although it is not suitable for small faces and low-quality images. The presented experimental results confirmed that the face recognition rate is strongly dependent on the detection precision. Using our method face recognition can be improved significantly, and it may be expected that alternative advanced face analysis biometric systems would benefit from the increased precision as well.

There is still a considerable difference between the face recognition rate obtained using the eye positions determined by human observers and those detected automatically. This means that in face recognition there is a need for further improvement of the detection precision. Possible directions of future works include investigating wavelet analysis of eye area, as well as eyelids detection to further increase the precision. Moreover, the detection time may be decreased by parallelization of the algorithm which will make it applicable to detection of small faces. Finally, the proposed method can be combined with alternative face detectors suitable for detecting small faces. The third level of our method can be used afterwards to increase the detection precision and improve advanced face analysis systems.

References

[1] M.-H. Yang, D. Kriegman, and N. Ahuja, "Detecting faces in images: A survey," *IEEE Trans. Pattern Anal. and Machine Intelligence*, vol. 24, pp. 34–58, 2002.

[2] W. Zhao, R. Chellappa, P. Phillips, and A. Rosenfeld, "Face recognition: A literature survey," *ACM Comput. Surv.*, vol. 35, no. 4, pp. 399–458, 2003.

[3] P. Phillips, P. Flynn, T. Scruggs, K. Bowyer, J. Chang, K. Hoffman, J. Marques, J. Min, and W. Worek, "Overview of the Face Recognition Grand Challenge," in *Proc. IEEE Conf. Computer Vision and Pattern Recogn.*, vol. 1, pp. 947–954, June 2005.

[4] M. Kawulok, J. Wu, and E. R. Hancock, "Supervised relevance maps for increasing the distinctiveness of facial images," *Pattern Recogn.*, vol. 44, no. 4, pp. 929–939, 2011.

[5] H. Rowley, S. Baluja, and T. Kanade, "Neural network-based face detection," vol. 20, no. 1, pp. 23–38, 1998.

[6] D. Maio and D. Maltoni, "Real-time face location on gray-scale static images," *Pattern Recogn.*, vol. 33, pp. 1525–1539, 2000.

[7] G. Yang and T. Huang, "Human face detection in complex background," *Pattern Recogn.*, vol. 27, 1994.

[8] C.-C. Han, H.-Y. Liao, K.-C. Yu, and L.-H. Chen, "Fast face detection via morphology-based pre-processing," in *Proc. Ninth Int. Conf. Image Anal. and Process.*, pp. 469–476, 1998.

[9] D. Ballard, "Generalizing the Hough transform to detect arbitrary shapes," *Pattern Recogn.*, vol. 13, pp. 111–122, 1981.

[10] R.-L. Hsu, M. Abdel-Mottaleb, and A. Jain, "Face detection in color images," *IEEE Trans. Pattern Anal. and Machine Intelligence*, vol. 24, pp. 696–706, May 2002.

[11] M. Jones and J. Rehg, "Statistical color models with application to skin detection," *Int. J. of Comp. Vision*, vol. 46, pp. 81–96, 2002.

[12] M. Turk and A. Pentland, "Face Recognition Using Eigenfaces," in *Proc. IEEE Conf. Computer Vision and Pattern Recogn.*, pp. 586–591, 1991.

[13] P. Viola and M. Jones, "Robust real-time face detection," *Int. J. of Computer Vision*, vol. 57, no. 2, pp. 137–154, 2004.

[14] R. Lienhart and J. Maydt, "An extended set of Haar-like features for rapid object detection," in *Proc. IEEE Int. Conf. on Image Process.*, pp. 900–903, 2002.

[15] J. Wu, S. Brubaker, M. Mullin, and J. Rehg, "Fast asymmetric learning for cascade face detection," *IEEE Trans. on Pattern Anal. and Machine Intelligence*, vol. 30, pp. 369–382, 2008.

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

[17] P. Wang and Q. Ji, "Multi-view face and eye detection using discriminant features," *Computer Vision and Image Underst.*, vol. 105, no. 2, pp. 99–111, 2007.

[18] C. Huang, H. Ai, Y. Li, and S. Lao, "High-performance rotation invariant multiview face detection," *IEEE Trans. on Pattern Anal. and Machine Intelligence*, vol. 29, pp. 671–686, 2007.

[19] Z. Li, L. Xue, and F. Tan, "Face detection in complex background based on skin color features and improved AdaBoost algorithms," in *Proc. IEEE Int. Conf. on Progress in Informatics and Computing (PIC)*, pp. 723 – 727, 2010.

[20] C. Wan, Y. Tian, H. Chen, and X. Wang, "Rapid face detection algorithm of color images under complex background," in *Proc. 8th Int. Symp. on Neural Networks, LNCS 6676*, pp. 356–363, 2011.

[21] W.-K. Tsao, A. J. T. Lee, Y.-H. Liu, T.-W. Chang, and H.-H. Lin, "A data mining approach to face detection," *Pattern Recogn.*, vol. 43, no. 3, pp. 1039–1049, 2010.

[22] D. W. Hansen and Q. Ji, "In the eye of the beholder: A survey of models for eyes and gaze," *IEEE Trans. on Pattern Anal. and Machine Intelligence*, vol. 32, pp. 478–500, 2010.

[23] R. Valenti and T. Gevers, "Accurate eye center location and tracking using isophote curvature," in *Proc. IEEE Conf. on Computer Vision and Pattern Recogn.*, 2008.

[24] J. Daugman, "The importance of being random: statistical principles of iris recognition," *Pattern Recogn.*, vol. 36, pp. 279–291, 2003.

[25] J. Ren and X. Jiang, "Fast eye localization based on pixel differences," in *Proc. IEEE Int. Conf. on Image Process.*, pp. 2733–2736, 2009.

[26] T. D'Orazio, M. Leo, G. Cicirelli, and A. Distante, "An algorithm for real time eye detection in face images," in *Proc. Int. Conf. on Pattern Recogn.*, pp. 278–281, 2004.

[27] Z. Zhu, K. Fujimura, and Q. Ji, "Real-time eye detection and tracking under various light conditions," in *ETRA*, pp. 139–144, 2002.

[28] P. Hillman and J. Hannah, "Global fitting of a facial model to facial features for model-based video coding," in *Proc. Third Int'l Symp. Image and Signal Process. and Anal.*, vol. 1, pp. 359–364, 2003.

[29] M. Castrillón-Santana, O. Déniz Suárez, M. Hernández Tejera, and C. Guerra Artal, "ENCARA2: Real-time detection of multiple faces at

different resolutions in video streams," *J. of Visual Communic. and Image Represent.*, pp. 130–140, April 2007.

[30] D. W. Hansen and J. P. Hansen, "Robustifying eye interaction," in *Proc. IEEE Computer Vision and Pattern Recogn. Workshop*, p. 152, 2006.

[31] M. Castrillón-Santana, D. Hernández-Sosa, and J. Lorenzo-Navarro, "Viola-Jones based detectors: How much affects the training set?," in *IbPRIA, LNCS 6669*, pp. 297–304, 2011.

[32] C. Cortes and V. Vapnik, "Support-Vector Networks," *Machine Learning*, vol. 20, no. 3, pp. 273–297, 1995.

[33] O. Jesorsky, K. J. Kirchberg, and R. Frischholz, "Robust face detection using the Hausdorff distance," in *Proc. 3rd Int. Conf. on Audio- and Video-Based Biometric Person Authentication*, pp. 90–95, Springer-Verlag, 2001.

[34] P. Phillips, H. Wechsler, J. Huang, and P. Rauss, "The FERET database and evaluation procedure for face recognition algorithms," *Image and Vision Computing J.*, vol. 16, no. 5, pp. 295–306, 1998.

[35] A. Martinez and R. Benavente, "The AR face database," Tech. Rep. 24, CVC, 1998.

[36] P. Belhumeur, J. Hespanha, and D. Kriegman, "Eigenfaces vs. Fisherfaces: recognition using class specific linear projection," *IEEE Trans. Pattern Anal. and Machine Intelligence*, vol. 19, pp. 711–720, July 1997.