Available online at www.sciencedirect.com

ScienceDirect

J. Vis. Commun. Image R. 18 (2007) 130–140

JOURNAL OF VISUAL Communication & IMAGE Representation

www.elsevier.com/locate/jvci

ENCARA2: Real-time detection of multiple faces at different resolutions in video streams

M. Castrillón *, O. Déniz, C. Guerra, M. Hernández

Instituto Universitario de Sistemas Inteligentes y Aplicaciones Numéricas en Ingeniería (IUSIANI), Universidad de Las Palmas de Gran Canaria, Las Palmas de Gran Canaria 35017, Spain

Received 10 June 2005; accepted 15 November 2006
Available online 22 January 2007

Abstract

This paper describes a face detection system which goes beyond traditional face detection approaches normally designed for still images. The system described in this paper has been designed taking into account the temporal coherence contained in a video stream in order to build a robust detector. Multiple and real-time detection is achieved by means of cue combination. The resulting system builds a feature based model for each detected face, and searches them using the various model information in the next frame. The experiments have been focused on video streams, where our system can actually exploit the benefits of the temporal coherence integration. The results achieved for video stream processing outperform Rowley–Kanade's and Viola–Jones' solutions providing eye and face data in real-time with a notable correct detection rate, approx. 99.9% faces and 87.5% eye pairs on 26338 images.

Keywords: Face detection; Real-time; Human–computer interaction

1. Introduction

Computer vision based people detection should be a basic ability to include in any Vision Based Interface [1]. Several approaches have been developed in the past for people detection attending to different elements of the human body: the face [2,3], the head [4,5], the entire body [6] or just the legs [7], as well as the human skin [8].

Among those body parts, the face plays a critical role in human communication [9]. Indeed, there are different static and dynamic features that we use to successfully interact with other people and to identify them. In this sense, if Human–Computer Interaction (HCI) could be more similar to human communication, HCI would be non-intrusive, more natural and comfortable for humans [10]. As mentioned above, in this context the face is a main information channel, and therefore our effort in this work has been focused on its detection, in order to build a data provider for face analyzers.

Face detection is a revisited topic in the literature with recent successful results [11–13]. However, these detectors focus on the problem using approaches which are valid for restricted face dimensions and, with the exception of the first reference, to a reduced head pose range.

In this paper, we describe a real-time vision system which goes beyond traditional still image face detectors, adding to a state of the art object centered face detector [13] elements in order to get a better, more robust, more flexible and real-time multiresolution face detector. The additions are related to: (1) the integration of knowledge about features, particularly eye location is also provided, present in faces, (2) the integration of the temporal coherence, (3) and the advantages evidenced by the local context in head detection for low resolution and difficult head poses [14]. These abilities extend the application of standard face detection systems, building a system which is able to manage robustly not only typical desktop interactions

* Corresponding author. Fax: +34928458711.
E-mail address: mcastrillon@iusiani.ulpgc.es (M. Castrillón).

1047-3203/$ - see front matter © 2006 Elsevier Inc. All rights reserved.
doi:10.1016/j.jvcir.2006.11.004

but also surveillance situations and the transition between both contexts, i.e. face and head detection.

2. Face detection

The standard face detection problem, given an arbitrary image, can be defined as: *to determine any face -if any- in the image returning the location and extent of each* [2,3]. Ideally, the whole procedure must perform in a robust manner for illumination, scale and orientation changes in the subject. Thus, robustness is a main aspect that must be taken into account by any face detector developer.

Face detection methods can be classified according to different criteria. In this paper, we have considered the information used to model faces to classify the different face detection techniques into two main families:

- Implicit or Pattern based: These approaches work searching exhaustively a previously learned pattern at every position and different scales of the input image, see Fig. 1.
- Explicit or Knowledge based: These approaches increase processing speed by taking into account face knowledge explicitly and combining cues such as color, motion and facial geometry and appearance.

Among the different approaches described in the literature, those belonging to the first family tackle the general problem of face detection in still images achieving great performance (and fast in recent developments) for the datasets available [2,3]. On the other hand, the techniques included in the second family provide faster performance, but only in restricted scenarios [2,3].

However, the problem of real-time face detection in the context of video streaming has not been properly focused. The direct application of typical face detectors to video streams neglects the integration of information which is implicit in the temporal behavior of the real sequence. As an example, this direct application will analyze the frame as if it were a still image, forgetting information provided by previous detections such as the position, size and appearance of the face detected.

Therefore, the approach described in this paper makes use of elements of both families trying to get their advantages, i.e., high performance given by the first family, and speed provided by the second family. Our approach integrates the temporal coherence in the system, as it is designed to exploit it during video processing. The integration of other cues help to improve the final system performance and robustness.

For comparison purposes we have chosen two well-known approaches from the first family, Rowley–Kanade's [15] and Viola–Jones' [13] detectors which are described briefly below. Both approaches are available for comparison purposes, and they also provide high detection performance, but particularly the second approach is able to perform almost at frame rate.

The reason to avoid any explicit based approach for comparison purposes is due to the fact that implicit based detectors provide better performance. Indeed our first face detector, called ENCARA, was a detector based on skin color model [16], it could perform twice faster than Viola–Jones' detector, but the reliability was reduced to specific lighting conditions. That was not a new result, indeed skin color based approaches have the lack of robustness for different conditions. A well known problem is the absence of a general skin color representation for any kind of light source and camera [17]. However, if a skin color approach is combined with an implicit based approach, this restriction can be avoided. This combinational paradigm is taken into consideration in our detector.

2.1. Rowley–Kanade's detector

Rowley–Kanade's detector [15] uses a multilayer neural network trained with multiple face and non-face prototypes at different scales, considering faces in almost upright position. The use of non-face appearance allowed to describe better the boundaries of the facial class.

Comparative results seem to improve those achieved previously by [18]. The system assumes a range of working sizes (starting at 20 × 20) as it performs a multiscale search on the image. The system allows the configuration of its tolerance for lateral views.

The process is computationally expensive and some optimization would be desirable to reduce the processing time. According to the authors [15], a fast version of the system can process a 320 × 240 pixel image in two to four seconds on a 200 MHz R4400 SGI Indigo 2. They also pointed out that color information, if available, may be

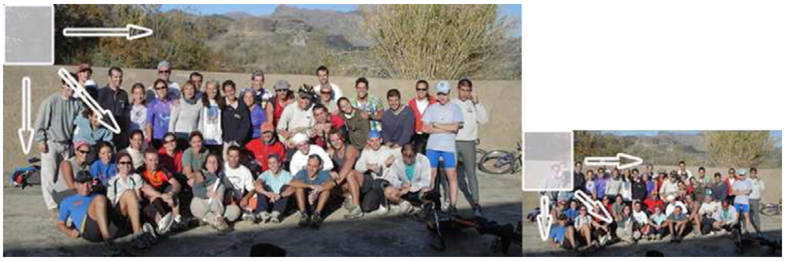

Fig. 1. The implicit based approaches shift the matching window on the image at different resolutions.

used to optimize the algorithm by means of restricting the search area, therefore improving performance.

2.2. Viola–Jones's detector

Recent implicit face detectors [12,13] have reduced dramatically the processing latency at high levels of accuracy. Particularly the general object detector framework described in [13], designed for rapid object detection, is based on the idea of a boosted cascade of weak classifiers. For each stage in the cascade, see Fig. 2, a separate subclassifier is trained to detect almost all target objects while rejecting a certain fraction of the non-object patterns (which were accepted by previous stages).

The resulting detection rate, D, and the false positive rate, F, of the cascade is given by the combination of each single stage classifier rates:

$$D = \prod_{i=l}^{K} d_i \quad F = \prod_{i=l}^{K} f_i \tag{1}$$

Each stage classifier is selected considering a combination of features which are computed on the integral image, see Fig. 3a. These features are reminiscent of Haar wavelets and early features of the human visual pathway such as center-surround and directional responses, see Fig. 3b. The implementation [19] integrated in the OpenCV (Open Computer Vision Library) [20] extends the original feature set [13]. As an example, the features achieved for the first stage of, respectively, a frontal face detector, and a head and shoulders detector are presented in Fig. 4. Both detectors are integrated in recent OpenCV releases [20].

Under this approach, given a 20 stage detector designed for refusing at each stage 50% of the non-object patterns (target false positive rate) while falsely eliminating only 0.1% of the object patterns (target detection rate), its expected overall detection rate is $0.999^{20} \sim 0.98$ with a false positive rate of $0.5^{20} \approx 0.9 * 10^{-6}$. This schema allows a high image processing rate, due to the fact that background regions of the image are quickly discarded while spending more time on promising object-like regions. Thus, the detector designer chooses the desired number of stages, the target false positive rate and the target detection rate per stage, achieving a tradeoff between accuracy and speed for the resulting classifier.

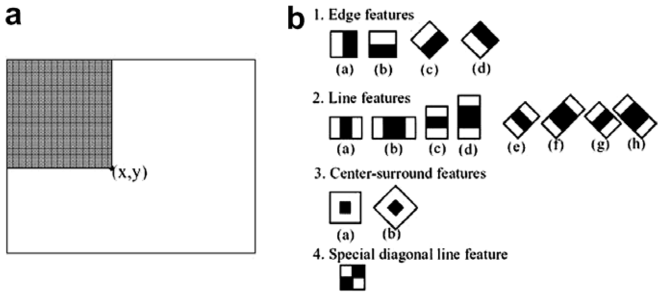
a
(x,y)
b
1. Edge features
(a) (b) (c) (d)
2. Line features
(a) (b) (c) (d) (e) (f) (g) (h)
3. Center-surround features
(a) (b)
4. Special diagonal line feature

Fig. 3. (a) The Integral Image stores integrals over subregions of the image, (b) features prototypes considered in [19] implementation.

3. Our face detection approach: ENCARA2

As mentioned above, our approach is related to both categories described in the previous section, as it makes use of both implicit and explicit knowledge to get the best of each one in an opportunistic fashion. The explicit knowledge is based on the face geometry and the descriptors extracted from a detection: color and appearance. On the other side, the implicit knowledge is integrated using the general object detection framework [13] which combines increasingly more complex classifiers in a cascade. The focus is extended for real-time modelling each detected face. Therefore this information is used based on temporal coherence to speed up the next frame processing.

3.1. The face detection loop procedure

The process used to face detection, see Fig. 5 for a schematic description, has two different working modes depending on recent face detection events reported:

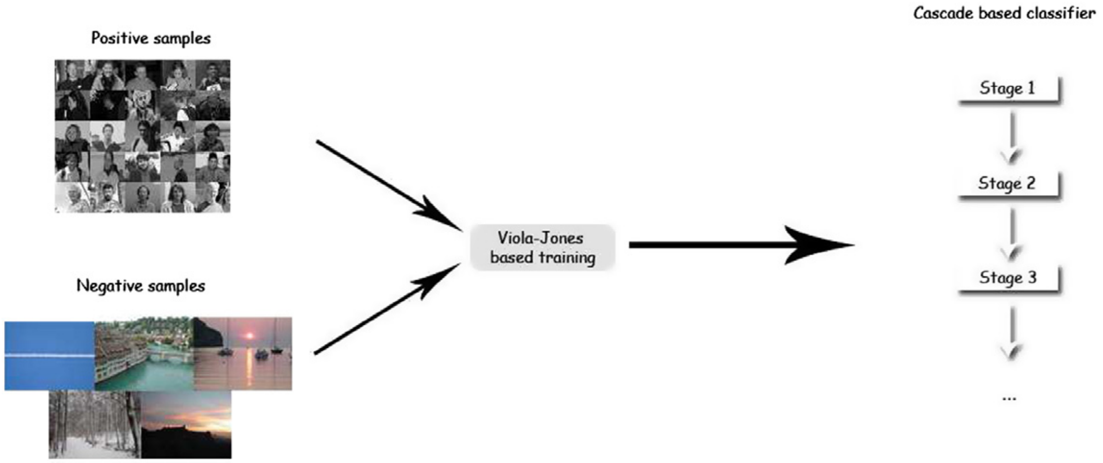

Fig. 2. Typical training procedure for a Viola–Jones' based classifier. Each stage classifier is obtained using positive and negative samples accepted by the previous stage.

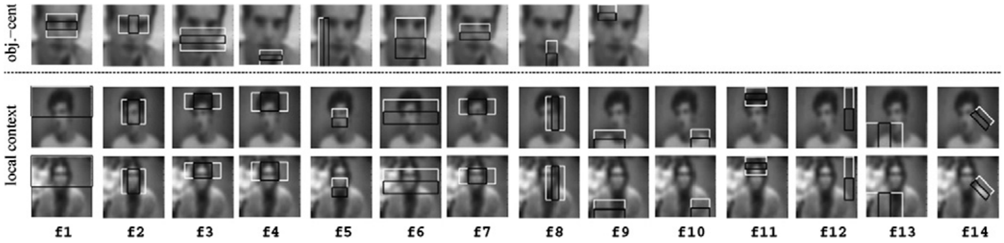

Fig. 4. Automatically extracted features of the first stage for frontal face (object centered) and head and shoulders (local context) detection respectively (extracted from [14]).

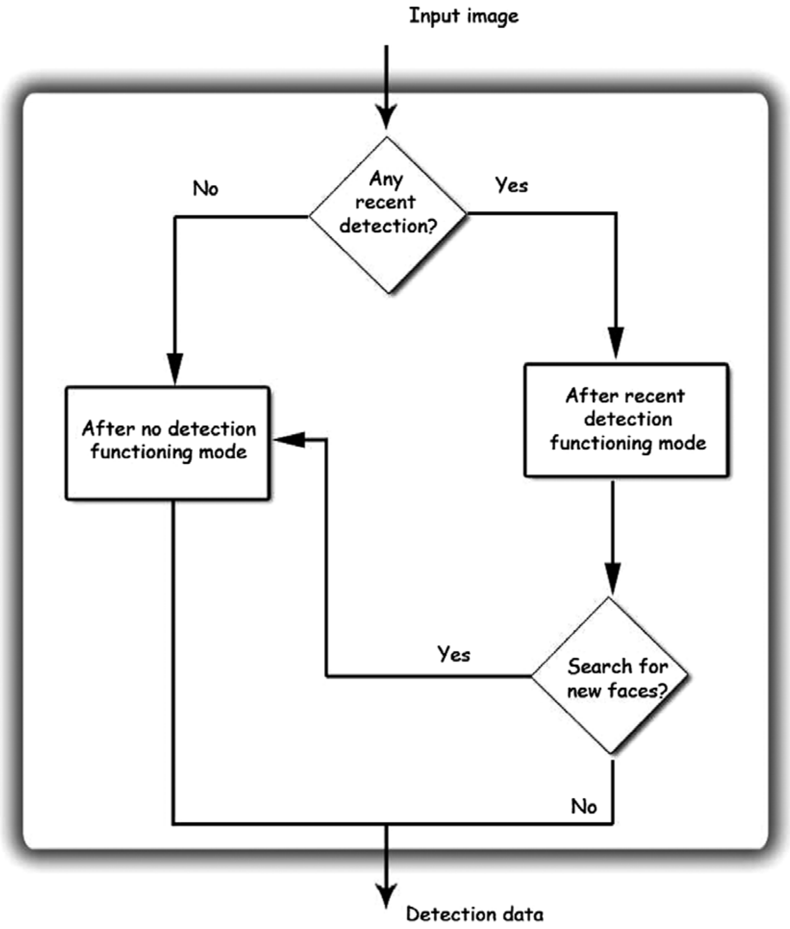

Fig. 5. ENCARA2 main modules.

After no detection: This working mode, see Fig. 6 for an overview, takes place at the beginning of an interaction session, when all the individuals are gone from the field of view, or if nobody is detected for a while. The approach basically makes use of two window shift detectors based on the general object detection framework described in [13]. These two brute force detectors are the frontal face detector described in [13], and the local context based face detector described in [14]. The last one achieves better recognition rates for low resolution images if the head and shoulders are visible. In order not to waste processing time, see Fig. 1 to understand their processing cost, the detectors are executed alternatively, i.e. one is applied to odd and the other to even frames.

Faces or head and shoulders smaller than the minimum pattern size, respectively 24 × 24 and 20 × 20 pixels, will not be located by the detector. Whenever a face or head is detected, the system models its color from the face/head container. Then it uses that modelled color trying to detect the facial features assuming that it is a frontal face, and therefore they would verify some geometric restrictions. The current implementation searches only the eyes using different alternatives for eye detection as described in detail in Section 3.2.

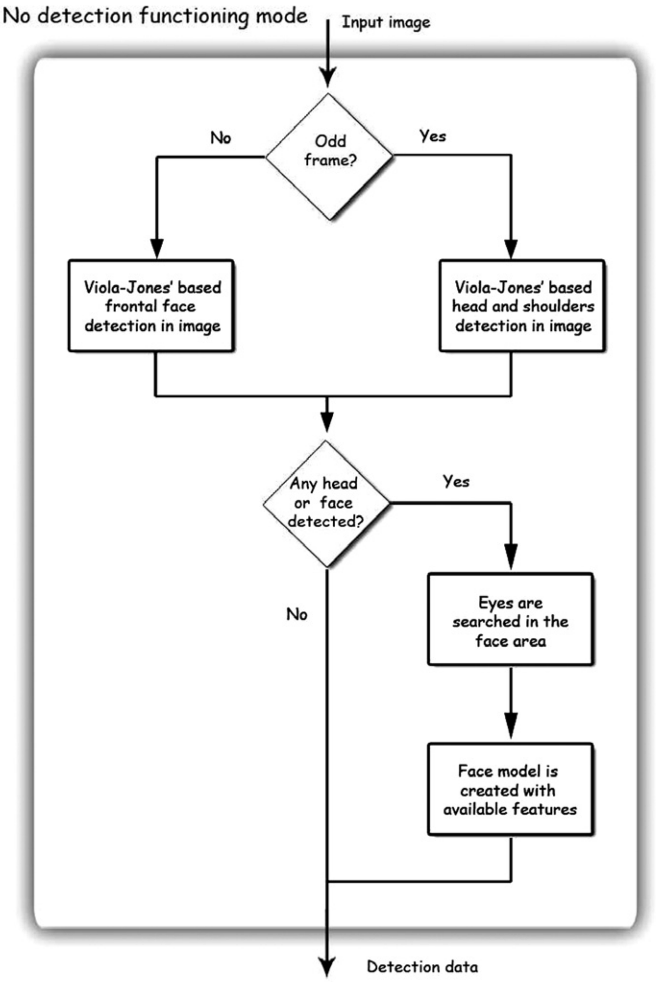
No detection functioning mode
Input image
Odd frame?
No
Yes
Viola-Jones' based frontal face detection in image
Viola-Jones' based head and shoulders detection in image
Any head or face detected?
Yes
No
Eyes are searched in the face area
Face model is created with available features
Detection data

Fig. 6. No recent detection working mode.

Finally, for each detected face, the system stores not only its position and size, but also its average color using red-green normalized color space [21] (considering just the center of the estimated face container provided by any of Viola-Jones based detectors), and the patterns of the eyes (if detected) and the whole face. Thus, a face is characterized by $f = \langle pos, size, red, green, leye_{pos}, leye_{pattern}, reye_{pos}, reye_{pattern}, face_{pattern}\rangle$.

After recent detection(s): In this working mode, see Fig. 7 for a graphical overview, each face detected has been modelled using different features. These features direct different cues which are applied opportunistically in the new image in an order based on their computational cost and reliability. These techniques are used to redetect a face, thus they are focused in a sub window of the image as expressed in Fig. 8. It must also be observed that these techniques are applied until one of them finds a new face coherent with the previous detection, therefore, their execution is not necessary for every frame. These considerations will speed up the whole process.

- Eye tracking: If eye patterns are available in the face model, a fast tracking algorithm [22] searches the minimum difference in the search area as follows:

$$D(u,v) = \sum_{\text{Area}} |I(u+i, v+j) - P(i,j)| \quad (2)$$

A dynamically updated threshold is used to decide if the eyes have been lost or not [22].

- Frontal face detector: A Viola–Jones' based face detector [13] will be used applied in the search window only if the tracker does not track the eyes.
- Local context face detector: If previous techniques fail, the local context based face detector [14] is applied in the search area.
- Skin color: If previous cues fail, the modelled skin color is used to locate the face in the search area. If a proper blob is located, eyes will be searched, see details in Section 3.2.
- Face tracking: If everything else fails, the prerecorded face pattern is tracked [22] in the search area. However, the tracking is not allowed to be the only valid cue for more than some consecutive frames in order to avoid tracking problems. Instead, the other cues should confirm, from time to time, the human presence or the person will be considered lost.

Whenever a face is detected, and its eyes were not tracked, the skin color is used for facial features detection as detailed in Section 3.2.

Additionally, every fifth frame one of Viola–Jones' based detectors is applied to the whole image in order to detect new faces. Those new faces are compared with those already detected by temporal coherence, removing the redundant ones. If no faces are detected for a while, the detector switches to the default *After no detection* working mode.

3.2. Eye detection

The process employed to detect the eyes assumes that the face detected is a frontal face. Therefore, it could happen that ENCARA2 will not provide eye locations for every detected face. This situation happens whenever the system fails detecting them or if both eyes are not visible. The eye pair detection process, graphically summarized in Fig. 9, is as follows:

(1) *Skin blob detection:* The skin color modelled is used to detect the face blob boundaries. The system heuristically removes elements that are not part of the face, e.g. neck, and fits an ellipse to the blob in order to rotate it to a vertical position [23].

(2) *Eyes location:* Different alternatives are used to locate the eyes:

(a) *Dark areas:* Eyes are particularly darker than their surroundings [24].

(b) *Viola-Jones based eye detector:* As the eye position can be roughly estimated and therefore restricted, a Viola–Jones' based eye detector provides fast performance. The detector searches eyes with a

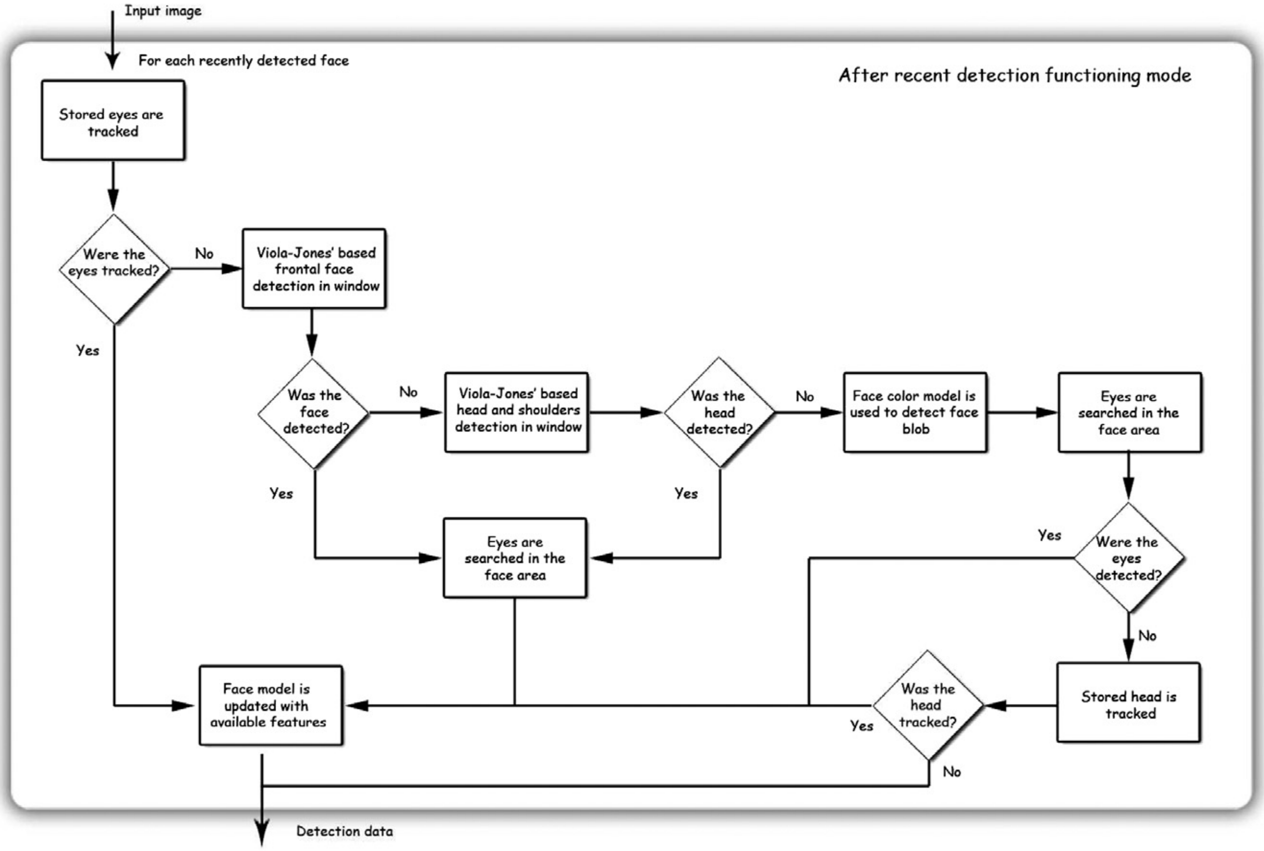
Input image
For each recently detected face
After recent detection functioning mode
Stored eyes are tracked
Were the eyes tracked?
No
Yes
Viola-Jones' based frontal face detection in window
Was the face detected?
No
Yes
Viola-Jones' based head and shoulders detection in window
Was the head detected?
No
Yes
Face color model is used to detect face blob
Eyes are searched in the face area
Were the eyes detected?
Yes
No
Eyes are searched in the face area
Stored head is tracked
Was the head tracked?
Yes
No
Face model is updated with available features
Detection data

Fig. 7. After recent detection working mode.

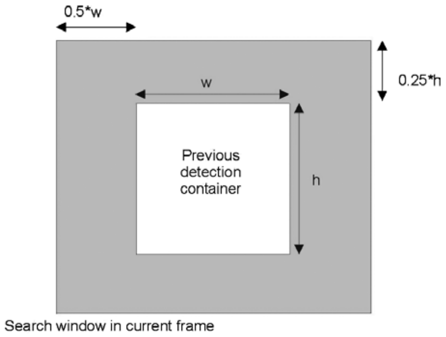

Fig. 8. The search area used for each detected face in the next frame is defined as an expansion of the previous face detection container.

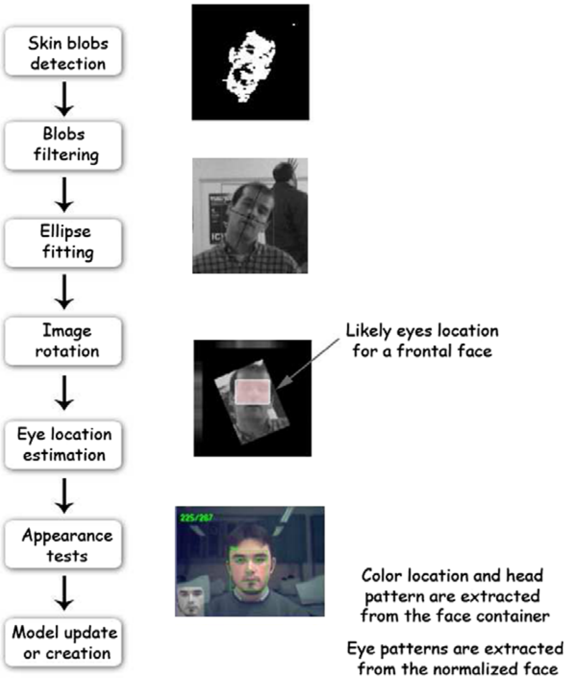
Skin blobs detection
Blobs filtering
Ellipse fitting
Image rotation
Eye location estimation
Appearance tests
Model update or creation
Likely eyes location for a frontal face
Color location and head pattern are extracted from the face container
Eye patterns are extracted from the normalized face

Fig. 9. Eye detection process.

minimum size of 16 × 12 pixels. For small faces, they are scaled up before performing the search.

(c) *Viola-Jones based eye pair detector:* If other cues fail, the eye pair detection can provide another estimation for eye positions in order to apply again steps (a and b). The minimum pattern size searched is 22 × 5.

(3) *Normalization:* Eye positions, if detected, provide a measure to normalize the frontal face candidate to a standard size.

(4) *Pattern Matching Confirmation:* Once the likely face has been normalized, its appearance is checked in two steps making use of Principal Component Analysis (PCA) [25]. Two PCA spaces were built using a face dataset of 4000 facial images extracted from internet and annotated by hand.

(a) *Eye appearance test:* A certain area (11 × 11) around both eyes in the normalized image is projected to a PCA space and reconstructed. The reconstruction error [26] provides a measure of its eye appearance, and can be used to identify incorrect eye detections.

(b) *Face appearance test:* A final appearance test applied to the whole normalized image. The image is first projected to a PCA space, and later its appearance is tested using a Support Vector Machine (SVM) classifier [27].

3.3. Multiple face detection: Detection threads

The approach considers the possibility of multiple face detection, as no restriction is imposed in that sense. As mentioned above, each face detected is described using some features, which serve for video streams to relate the detection information achieved in consecutive frames, especially when multiple individuals are present. During the video stream processing, the face detector gathers a set of *detection threads*, $IS = \{\mathrm{d}t_1, \mathrm{d}t_2, \ldots, \mathrm{d}t_n\}$. A detection thread contains a set of continuous detections, i.e. detections which take place in different frames but are related by the system in terms of coherence of position, size and pattern matching. Thus, for each detection thread, the face detector system provides a number of facial samples, $\mathrm{d}t_p = \{x_1, \ldots, x_{m_p}\}$, which correspond to those detections for which also the eyes were located.

Viola–Jones' based detectors have some level of false detections. For that reason a new detection thread is created only after the eyes have been also detected. The use of color and tracking cues after a recent detection is reserved to detections which are already considered part of a detection thread. In this way, spurious detections do not launch cues which are not robust enough, in the sense that they are not able to recover from a false face detection.

4. Experiments

4.1. Static images

ENCARA2 has not been designed to improve still images detection with the exception of providing additional eye locations. Indeed due to the fact that no temporal coherence can be used, its performance in that context combines the results achieved for the standard Viola–Jones' face detector [13] and the local context based face detector [14]. We forward the reader to those works to get precise information for static images results. In any case, we would like to present some results in Fig. 10, to clarify the different detection levels that ENCARA2 provides. Three different kinds of detections are possible: (a) Pure Viola–Jones' based frontal face detections (white containers), (b) frontal faces whose eyes were also detected by means of additional color processing (gray containers), and (c) Viola–Jones' based head and shoulders detection (two concentric containers).

4.2. Video streams: Desktop scenarios

The strength of our approach is mainly exploited in video stream processing thanks to cue integration. Seventy-four sequences corresponding to different individuals, cameras and environments with a resolution of 320 × 240 were recorded and processed. The results described in Table 1 describe the performance achieved processing sequences which present a single individual sat and speaking in front of the computer or moderating a TV news program, see Fig. 11 for some samples. Therefore, the face pose is mainly frontal, but it is not controlled, i.e. lateral views and occlusions due to arm movements are possible. Therefore the eyes are not always visible. The total set contains 26,338 images, presenting all of them a single face easily detected by a human.

In order to check the detectors performance, the sequences have been manually annotated, therefore the face containers are available for the whole set of images. However, eye locations are available only for a subset of 4059 images. The eyes location allows us to compute the actual distance between them, which will be referred below as *EyeDist*. This value will be used to estimate the goodness of eye detection.

Two different criteria have been defined to establish whether a detection is correct:

Correct face criterium: A face is considered correctly detected, if the detected face overlaps at least 80% of the annotated area, and the area difference is not doubled.

Correct face criterium: The eyes of a face detected are considered correctly detected if for both eyes the distance to manually marked eyes is lower than a threshold that depends on the actual distance between the eyes, *EyeDist*. The threshold considered was *EyeDist*/4 similarly to [28]. The same authors confirm in [29], that their threshold is reasonable for further face analysis.

Table 1 presents the results obtained after processing the whole set of sequences with the different detectors, i.e. 26,338 images. The correct detection ratios (TD) are given considering the whole sequence, and the false detection ratios (FD) are related to the total number of detections. Rowley's detector is notably slower than the others, but it provides eye detection for the 78% of detected faces, feature which is not considered by Viola–Jones' detector. As for our detector, it is observed that it performs more than twice faster than Viola–Jones' detector, and almost ten times faster than Rowley's. This performance is accompanied by a number of correct detections for faces and eyes which is always greater, in absolute value, than any of the other two approaches. It is observed that eye detection reflects a larger improvement in comparison to Rowley's detector. False detections are in many cases associated to detections which have not been properly sized.

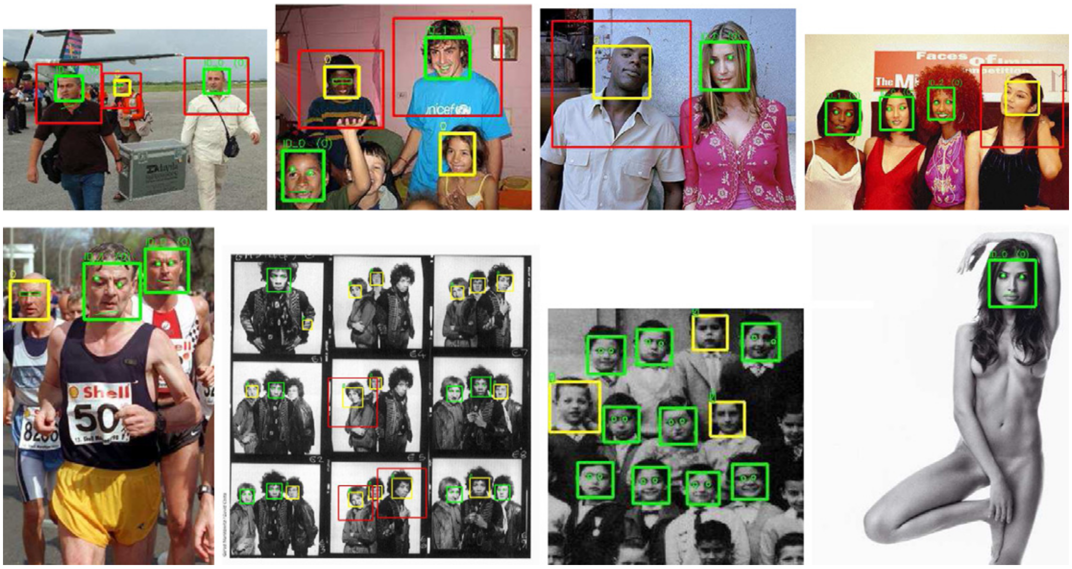

Fig. 10. Detection examples for some CMU database samples [12] and images extracted from Internet.

Table 1
Results for face and eye detection processing using a PIV 2.2 Ghz

	Rowley		Viola		Our detector	
	TD (%)	FD (%)	TD (%)	FD (%)	TD (%)	FD (%)
Faces	89.27	2.16	97.69	8.25	99.92	8.07
Left eye	77.51	0.8	0.0	—	91.83	4.04
Right eye	78.18	1	0.0	—	92.48	3.33
Proc. time	422.4 ms		117.5 ms		45.6 ms	

Fig. 11. Sample sequences.

In at least 10 of the sequences there were detections which correspond to non face patterns (provided by Viola–Jones' detectors). However these detections were correctly not assigned to any detection thread as the eyes were not found and their position, color and size were not coherent with any active detection thread.

Only for three sequences with a single individual, the detection thread was not unique. This means that the system could not consider as continuous the presence of the individual in the video stream. In these sequences this was due to the fact that at a certain point a detection thread was incorrectly fused with an erroneous detection in the current frame. However, in all the cases the detection thread was shortly considered lost, and therefore some frames later the still present face was newly detected, and a new detection thread created. This is a really interesting result considering the large changes in pose experimented in many of the sequences.

4.3. Cue integration benefits

The integration of different cues is exemplified in Fig. 12. In that Figure, detections depicted were not provided by Viola–Jones' based detectors: (a) Grey squared faces have been detected using eye tracking, (b) dark ones by means of head tracking (the last possibility), and (c) white faces by means of color detection. Indeed after an initial detection, the other cues were able to manage the pose changes without losing the face/head. It must be observed that eyes are located only for frontal poses in the current implementation.

Cue combination helps the system to fit the real-time restriction. Both Rowley's and Viola–Jones' detectors perform an exhaustive search in the image at different scales, see Fig. 1. Each approach employs a different technique for matching but in any case they depend on the image resolution and the number of scales considered typically the

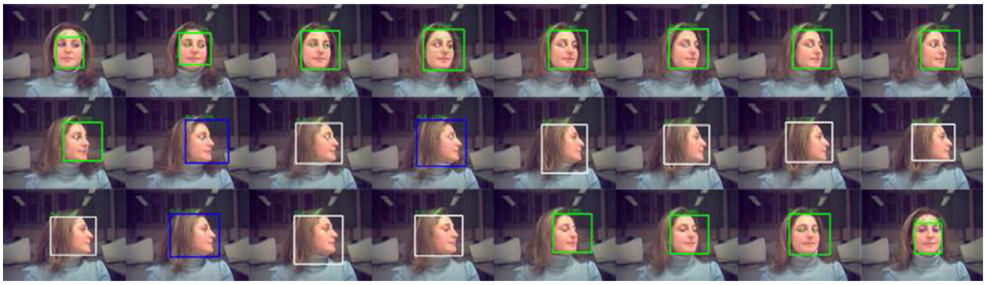

Fig. 12. Pose changes can be managed by means of cue combination.

image is repeatedly downsampled 1.1 times until a minimal size is reached, see Fig. 1. Thus, the processing cost per frame:

$$\mathit{Viola\_Cost} = o(\mathit{Single\_Viola\_matching\_cost} \times \mathit{width} \times \mathit{height} \times \mathit{nscales}) \quad (3)$$

Table 1 evidences that the cost of a single matching operation is greater using Rowley's approach. For that reason we have chosen Viola–Jones' approach as the implicit basement for our development. ENCARA2 in the worst case, if no faces have been detected, will behave similarly to a standard Viola–Jones' detector. But whenever there is a detection the cost per frame will be modified:

$$\begin{aligned}&\mathit{ENCARA2\_Cost}\\ &\quad = O(\mathit{nfacesdetected} \times \mathit{Eyestrackingcost}\\ &\qquad + \mathit{ViolaA\_Cost\_in\_window} + \mathit{ViolaB\_Cost\_in\_window}\\ &\qquad + \mathit{Color\_based\_cost} + \mathit{Head\_tracking\_Cost}) \quad (4)\end{aligned}$$

Observe that this cost is again the worst case, which happens when no cue is able to redetect a face. In the desktop context considered in the experiments that worst case is typically not present. Indeed the value reflected in 4 is in general lower if the other cues integrated: Tracking, Color and Subwindow detection, are able to detect. Therefore, every frame does not require all the processing just till the face is again detected.

$$\begin{aligned}\mathit{Eye\_stracking\_cost} &= O(2 \times \mathit{Eye\_matching\_cost} \times \mathit{subwindow\_width}\\ &\quad \times \mathit{subwindow\_height}) \quad (5)\\ \mathit{Viola\_Cost\_in\_window} &= O(\mathit{Single\_Viola\_matching\_cost} \times \mathit{window\_width}\\ &\quad \times \mathit{window\_height} \times \mathit{nscales}) \quad (6)\\ \mathit{Color\_based\_cost} &= O(\mathit{window\_width} \times \mathit{window\_height}\\ &\quad + \mathit{Eye\_detection\_cost}) \quad (7)\\ \mathit{Head\_tracking\_cost} &= O(\mathit{Head\_matching\_cost}\\ &\quad \times \mathit{window\_width} \times \mathit{window\_height}) \quad (8)\end{aligned}$$

The results described in Table 1, show that cue integration reduces for typical desktop scenarios sequences, the time consumption in 1/3, adding the possibility of eye detection in many frontal views.

For multiple individuals sequences, the system needs more time as more faces are tracked simultaneously, in our experiments around 20 ms. per individual added to the image. This effect can be reduced by decreasing the number of times per second that new faces are searched in the whole image.

A multiple face detection example is presented in Fig. 13. From left to right: (1) Both faces are detected and their eyes, (2) the Viola based detectors failed detecting the right face, it is detected by tracking the face pattern, (3) the left face is detected using skin color and the right one by means of the local context face detector, (4) the same for the left face, the right one is found by tracking, (5) face pattern tracking is not allowed to be the only valid cue for many consecutive frames, so the right face detection thread is considered missed, and (6) the right face recovers its vertical position and is fused with the latent detection thread.

4.4. Video streams: Unrestricted scenarios

Preliminary experiments have been performed also for sequences which are not restricted to a desktop context. Some results achieved for detection at different resolutions can be observed in Figs. 14 and 15.

The face location for the sequence corresponding to Fig. 14 has been manually annotated. Table 2 presents the detection rates summary. For Viola–Jones' detector the detection rate hardly reaches 30%. This is due to the fact that the face is in many frames not frontal, and/or its resolution is reduced, situation which easily fools state of the art face detectors. Rowley's face detector would present the same problem. On the other hand the local context detector is able to get a better detection rate. Our

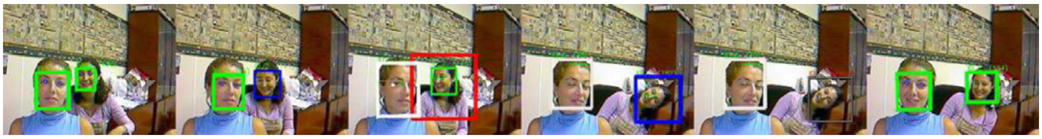

Fig. 13. Detection results in a test sequence with multiple individuals.

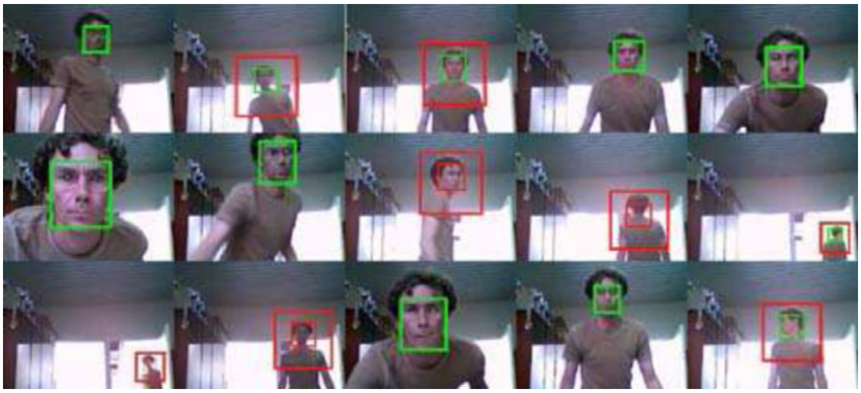

Fig. 14. Sample detections corresponding to an indoor sequence (320 × 240 pixels).

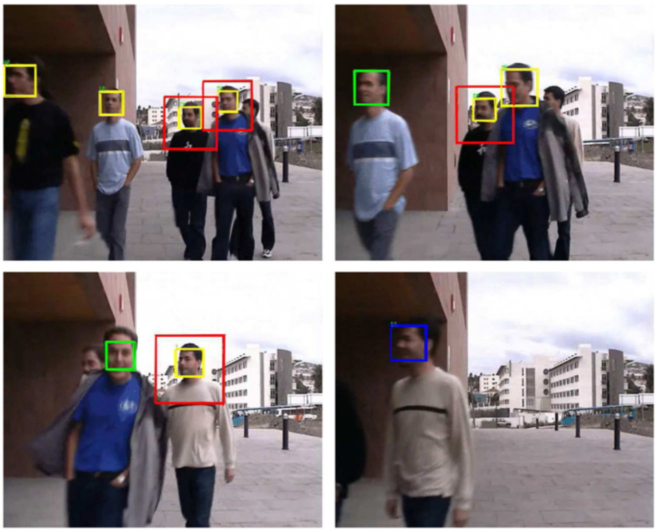

Fig. 15. Sample detections corresponding to an outdoor sequence (720 × 576 pixels).

Table 2
Results for the indoor sequence, see Fig. 14

Detector	Detection rate (%)	False detection rate (%)
Object centered [13]	30.5	0.0
Local context [14]	66	1.4
Our detector	81.8	0.3

system, which integrates both detectors added to the temporal coherence, outperforms clearly both approaches applied to a context closer to reality.

5. Conclusions and future work

We have presented an approach for face detection in video streams which makes use of a cascade combination in an opportunistic fashion of different classical face detection approaches for video stream, but integrating some elements of temporal coherence. Therefore, we pursue to integrate the benefits of both families of face detection approaches: the robustness of implicit approaches and the speed of explicit approaches. The final system provides faster and better detection rates outperforming well known face detection systems. Detection rates achieved, 99.9% faces and 97% eye pairs detected on 26338 images, reported an error rate of 8 and 4% according to different error detection criteria extracted from the literature.

Additionally, the system is able to detect multiple faces and their eyes providing for the experiments an average processing rate of 45.6 ms per frame which makes the system suitable for further processing in the field of perceptual user interfaces. A demo application and a library for comparison purposes are provided under request to the authors.

Future work will focus on the improvement of the color module, and the detection of additional facial and context features in order to provide more elements to manage an unrestricted individual performance. For example, the inclusion of the individual identity and/or his clothes color model will help in situations where different individuals present extreme poses and overlap.

Acknowledgments

Work partially funded by research projects Univ. of Las Palmas de Gran Canaria UNI2003/06, UNI2004/10, UNI2004/25 and UNI2005/28, Canary Islands Autonomous Government PI2003/160 and PI2003/165 and the Spanish Ministry of Education and Science and FEDER funds (TIN2004-07087).

References

[1] M. Turk, Computer vision in the interface, Communications of the ACM 47 (1) (2004) 61–67.

[2] E. Hjelmas, B.K. Low, Face detection: a survey, Computer Vision and Image Understanding 83 (3) (2001) 236–274.

[3] M.-H. Yang, D. Kriegman, N. Ahuja, Detecting faces in images: a survey, Transactions on Pattern Analysis and Machine Intelligence 24 (1) (2002) 34–58.

[4] S. Birchfield, Elliptical head tracking using intensity gradients and color histograms, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 1998, pp. 232–237.

[5] M.L. Cascia, S. Sclaroff, Fast, reliable head tracking under varying illumination: an approach based on registration of texture-mapped 3d models, IEEE Trans on Pattern Analysis and Machine Intelligence 22 (4) (2000) 322–336.

[6] P. Viola, M.J. Jones, D. Snow, Detecting pedestrians using patterns of motion and appearance, in: Proceedings of the International Conference on Computer Vision, vol. 2, 2003, pp. 734–741.

[7] C. Papageorgiou, M. Oren, T. Poggio, A general framework for object detection, in: Proceedings of the International Conference on Computer Vision, 1998, pp. 555–562.

[8] M.J. Jones, J.M. Rehg, Statistical Color Models with Application to Skin Detection, Technical Report Series CRL 98/11, Cambridge Research Laboratory (December 1998).

[9] C.L. Lisetti, D.J. Schiano, Automatic facial expression interpretation: Where human–computer interaction, artificial intelligence and cognitive science intersect, Pragmatics and Cognition Special Issue on Facial Information Processing: A Multidisciplinary Perspective 8 (1) (2000) 185–235.

[10] A. Pentland, Looking at people: sensing for ubiquitous and wearable computing, IEEE Trans on Pattern Analysis and Machine Intelligence (2000) 107–119.

[11] S.Z. Li, L. Zhu, Z. Zhang, A. Blake, H. Zhang, H. Shum, Statistical learning of multi-view face detection, in: European Conference Computer Vision, 2002, pp. 67–81.

[12] H. Schneiderman, T. Kanade, A statistical method for 3d object detection applied to faces and cars, in: IEEE Conference on Computer Vision and Pattern Recognition, 2000, pp. 1746–1759.

[13] P. Viola, M.J. Jones, Robust real-time face detection, International Journal of Computer Vision 57 (2) (2004) 151–173.

[14] H. Kruppa, M. Castrillón Santana, B. Schiele, Fast and robust face finding via local context, in: Joint IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance (VS-PETS), 2003, pp.157–164.

[15] H.A. Rowley, S. Baluja, T. Kanade, Neural network-based face detection, IEEE Trans on Pattern Analysis and Machine Intelligence 20 (1) (1998) 23–38.

[16] M. Castrillón Santana, F. Hernández Tejera, J. Cabrera Gamez, Encara: real-time detection of frontal faces, in: International Conference on Image Processing, Barcelona, Spain, 2003.

[17] M. Storring, H.J. Andersen, E. Granum, Physics-based modelling of human skin colour under mixed illuminants, Robotics and Autonomous Systems.

[18] K.-K. Sung, T. Poggio, Example-based learning for view-based human face detection, IEEE Trans. on Pattern Analysis and Machine Intelligence 20 (1).

[19] R. Lienhart, A. Kuranov, V. Pisarevsky, Empirical analysis of detection cascades of boosted classifiers for rapid object detection, in: DAGM'03, Magdeburg, Germany, 2003, pp. 297–304.

[20] Intel, Intel open source computer vision library, b4.0, <www.intel.com/research/mrl/research/opencv> (August 2004).

[21] C. Wren, A. Azarrbayejani, T. Darrell, A. Pentland, Pfinder: Real-time tracking of the human body, IEEE Trans. on Pattern Analysis and Machine Intelligence 19 (7) (1997) 780–785.

[22] C. Guerra Artal, Contribuciones al seguimiento visual precategorico, Ph.D. thesis, Universidad de Las Palmas de Gran Canaria (Octubre 2002).

[23] K. Sobottka, I. Pitas, A novel method for automatic face segmentation, face feature extraction and tracking, Signal Processing: Image Communication 12 (3).

[24] S. Feyrer, A. Zell, Detection, tracking and pursuit of humans with autonomous mobile robot, in: Proceedings of the International Conference on Intelligent Robots and Systems, Kyongju, Korea, 1999, pp. 864–869.

[25] Y. Kirby, L. Sirovich, Application of the karhunen-love procedure for the characterization of human faces, IEEE Trans. on Pattern Analysis and Machine Intelligence 12(1).

[26] E. Hjelmas, I. Farup, Experimental comparison of face/non-face classifiers, in: Proceedings. of the Third International Conference on Audio- and Video-Based Person Authentication. Lecture Notes in Computer Science 2091, 2001, pp. 65–70.

[27] V. Vapnik, The Nature of Statistical Learning Theory, Springer, New York, 1995.

[28] O. Jesorsky, K.J. Kirchberg, R.W. Frischholz, Robust face detection using the hausdorff distance. Lecture notes in computer science, Proceedings of the Third International Conference on Audio- and Video-Based Person Authentication 2091 (2001) 90–95.

[29] K.J. Kirchberg, O. Jesorsky, R.W. Frischholz, Genetic model optimization for hausdorff distance-based face localization. Lecture notes in computer science, Biometric Authentication Person Authentication 2359 (2002) 103–111.