Real-time Detection of Faces in Video Streams

M. Castrillón-Santana, O. Déniz-Suárez, C. Guerra-Artal and M. Hernández-Tejera
IUSIANI - Edif. Ctral. del Parque Científico Tecnológico
Universidad de Las Palmas de Gran Canaria, Spain
mcastrillon@iusiani.ulpgc.es

Abstract

This paper describes a face detection system which goes beyond traditional approaches normally designed for still images. First the video stream context is considered to apply the detector, and therefore, the resulting system is designed taking into consideration a main feature available in a video stream, i.e. temporal coherence. 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 results achieved for video stream processing outperform Rowley-Kanade's and Viola-Jones' solutions providing eye and face data in a reduced time with a notable correct detection rate.

1 Introduction

People detection is a basic ability to be included in any Vision Based Interface [23] in order to use computer vision technology to perceive the user interacting in an Human Computer Interaction (HCI) context. Several approaches have been developed in the past for people detection attending to different elements of the human body: the face [7, 28], the head [1, 2], the entire body [26] or just the legs [16], as well as the human skin [10].

It is known that the human face plays a critical role in human communication [15]. 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 HCI could be more similar to human to human communication, HCI would be non-intrusive, more natural and comfortable for humans [17]. As mentioned above, in this context the face is a main information channel, and therefore first its detection and later its analysis must be accomplished.

Face detection is a revisited topic in the literature with recent successful results [13, 19, 25]. 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 [25] 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 present in faces, 2) the integration of the temporal coherence, 3) the advantages evidenced by the local context in head detection for low resolution and difficult head poses [12], and 4) facial elements location, particularly eyes in this implementation, are also provided. These abilities extend the application of standard face detection systems, building a system which is able to manage robustly not only typical desktop interactions 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* [7, 28]. 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 system.

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.
- 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 [7, 28]. On the other hand, the techniques included in the second family provide faster performance in restricted scenarios.

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 examples, we can point out the position, size and appearance of the face detected in the previous frame, instead of analyzing, as a standard face detector performs, the new frame forgetting the video stream history.

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. For comparison purposes we have chosen two well-known approaches from the first family, the Rowley-Kanade's [18] and the Viola-Jones' [25] detectors which are described briefly below. Both provide high detection performance, but particularly the second approach is able to perform almost at frame rate.

2.1 Rowley-Kanade's Detector

The Rowley-Kanade's detector [18] 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 described better the boundaries of the facial class.

Comparative results seem to improve those achieved previously by [22]. The system assumes a range of working sizes (starting at 20x20) 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. The authors warrant the reaching of responses in 2 - 4 seconds, on those days, when improving implementation, and also pointed out that color information, if available, may be used to optimize the algorithm by means of restricting the search area, therefore improving performance [18].

2.2 Viola-Jones's Detector

Recent implicit face detectors [19, 25] have reduced dramatically the processing latency at high levels of accuracy. Particularly the object detector framework described in [25], has been made available integrated in OpenCV (Open Computer Vision Library) [8]. This framework, designed for rapid object detection, is based on the idea of a boosted cascade of weak classifiers [25] but extends the original feature set and provides different boosting variants for learning [14].

The cascade learning algorithm is similar to decision-tree learning. Essentially, a classifier cascade can be seen as a degenerated decision tree. For each stage in the cascade a separate subclassifier is trained to detect almost all target objects while rejecting a certain fraction of the non-object patterns. 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:

$$D = \prod_{i=1}^{K} d_i \qquad F = \prod_{i=1}^{K} f_i \qquad (1)$$

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 $0.999^{20} \approx 0.98$ 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 trade-off between accuracy and speed for the resulting classifier.

3 Our Face Detection Approach

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 [25] 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 face detection, see Figure 1 for an schematic description, approach here described has two different working modes depending on recent face detection events reported:

After no detection: This working mode takes place at the beginning of an interaction session, when all the individuals are gone from the field of view, or if nobody is

0-7695-2319-6/05 $20.00 © 2005 IEEE

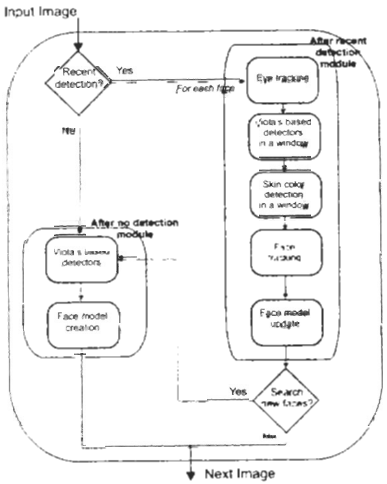

Figure 1: Face detector modules.

detected for a while. The approach basically makes use of two window shift detectors based on the general object detection framework described in [25]. These two brute force detectors, integrated in the last OpenCV release [8], are the frontal face detector described in that paper, and the local context based face detector described in [12]. The last one achieves better recognition rates for low resolution images if the head and shoulders are visible. The respective minimum size searched are 24×24 and 20×20 pixels. In order not to waste processing time, the detectors are executed alternatively.

For any face detected, the system tries to detect its facial features assuming that it is a frontal face, and therefore its facial features would verify some geometric restrictions. The current implementation searches only the eyes, using a process similar to the one employed in [3]. It has been however improved by the addition of different alternatives for eye detection as described below:

1. *Skin blob detection:* Once a face is detected, its skin color is modelled using red-green normalized color space [27], considering just the center of the estimated face container provided by any of the Viola-Jones based detectors. 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 [20].
2. *Eyes location:* At this point, the approach searches eye candidates in the likely areas inside the skin blob considering that the face detected is a frontal face. Different candidate pairs are checked for their appearance until one of them, is accepted. The cues used for this purpose are:
   (a) *Dark areas:* Eyes are particularly darker than their surroundings [4].
   (b) *Viola-Jones based eye detector:* As the eye position can be roughly estimated and therefore restricted, a Viola-Jones based eye detector provides very fast results. The detector searches eyes with a 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. The normalization step allows further face processing modules to reduce the problem dimensionality.
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) spaces [11]. The 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 [6] 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 [24].

After recent detection(s): As briefly mentioned above, for each detected face, the system stores not only its position and size, but also its average color using red-green normalized color space [27], and the patterns of the eyes (if detected) and the whole face. Thus, a face is characterized by $f = \langle pos, size, red, green, leye_{pos}, leye_{pattern}, reye_{pos}, reye_{pattern}, face_{pattern} \rangle$.

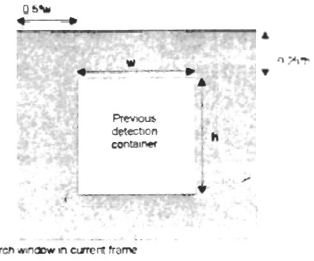

Figure 2: The search area used for each detected face in the next frame is defined as an expansion of the previous face detection container.

These features direct different cues in the next frames which are applied opportunistically in an order based on their computational cost and reliability. It must be noticed that these cues are not restricted to frontal faces, therefore the final detector system is more general.

- Eye tracking: A fast tracking algorithm [5] is applied in an area that surrounds previously detected eyes, if available. The tracker makes use of a fixed pattern size for both eyes, 24×24, and searches the minimum difference in the search area as follows:

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

  Eye patterns are previously saved with the first detection, and updated according to the strategies described in [5], i.e. only if there is a notorious change in relation to the original pattern, and this difference could confuse the tracker with any other pattern of the close context. If the difference reported is too hig, the pattern will be considered lost.
- Basic face detector: The Viola-Jones face detector [25] searches faces but only in an area that covers the previous detection, see Figure 2. This strategy significant reduces processing time.
- Local context face detector: If previous techniques fail, the local context based face detector is applied in an area that includes the previous detection [12], see Figure 2.
- Skin color: The integration of other cues, likely weaker, help to improve the final system performance and robustness. Skin color based approaches for face detection 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 [21]. However, the skin color extracted from the face previously detected by the Viola-Jones' detector can be used to estimate facial features position by means of the color blob, as described above. If previous cues fail, the modelled skin color is used to locate the face, and therefore it is searched in the window that contains the previous detection, see Figure 2. The new sizes and positions are coherently checked, due to the fact that the skin color container is not allowed to experiment large size changes just to avoid an incorrect color updating mechanism.
- Face tracking: If everything else fails, the prerecorded face pattern is searched in an area that covers previous detection [5], see Figure 2. The tracking pattern has a fixed size, for that reason the system scales down the face to fit it in the pattern size. The scale ratio is stored and later used if necessary to scale down the search area in the next frame. This action helps reducing the tracking shift problem. 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 the human presence, from time to time, or the person will be considered lost.

For each previous detection, these techniques are applied until one of them finds a new face coherent with the previous detection. Whenever a face is detected, and its eyes were not tracked, the skin color is used for facial features detection as explained above for the *After no detection* working mode. If the facial features were not located, likely for non frontal faces, the detection would be accepted if it matches with a coherent previous, in time, detection.

Also, every third frame one of the 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 and those which are redundant removed. If no faces are detected for a while, the process switches to the default *After no detection* working mode.

3.2 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*, $IS = \{dt_1, dt_2, ..., dt_n\}$. 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, $dt_p = \{x_1, ..., x_{ns_p}\}$, which correspond to those detections for which also the eyes were located.

The 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.

Ideally a detection thread contains samples detected of a single individual. However, different detection threads can correspond to the same individual, aspect which is not checked by the current implementation. Gaps are allowed during detection thread life, but a detection thread is considered lost if after a predefined number of frames it is not correctly related to a new detection.

4 Experiments

4.1 Static images

For static images the approach provides a performance which combines the results achieved for the standard Viola-Jones face detector [25] and the local context based face detector [12], see Figure 3 for some detection samples using the CMU database. We refer the reader to those works to get precise information for static images results.

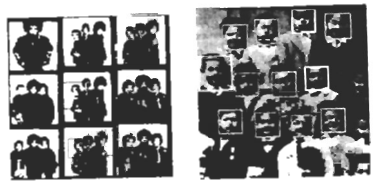

Figure 3: Detection examples for some CMU database samples [19]. A green square means that the eyes were detected, the yellow means that they were not detected (i.e. they are just Viola-Jones' detections), and the red containing a yellow rectangle means that the local context detector was used. The images have been scaled down to fit the paper size, their original sizes are 814×820, 256×256 respectively. For still images there are no detections based on color or tracking.

Figure 4: Sequences samples.

4.2 Video streams: Desktop scenarios

The strength of our approach is mainly exploited in video stream processing thanks to cue integration. 74 sequences, see Figure 4, corresponding to different individuals, cameras and environments with a resolution of 320×240 were recorded and processed. The total set contains 26338 images, presenting all of them a 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, and the eye locations are available for a subset of 4059 images.

Two different criteria have been defined to establish whether a detection is correct: 1) 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. 2) 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, $ground\_data\_inter\_eyes\_distance/4$ similarly to [9].

	Rowley		Viola		Our detector	
	TD	FD	TD	FD	TD	FD
Faces	89.2%	2.2%	97.7%	8.2%	99.9%	8%
Left Eye	77.51%	-	0.0%	-	91.8%	-
Right Eye	78.18%	-	0.0%	-	92.5%	-
Proc. time	422.4 msecs.		117.5 msecs.		45.6 msecs	

Table 1: Results for face and eye detection processing 26338 images. The correct detection ratios (TD) are given for the detections over the whole sequence, and the false detection ratios (TD) consider the total number of detections. (see Table 2 for error detection results related to eyes).

	Rowley		Our detector	
	TD	FD	TD	FD
Left Eye	67.7%	0.8%	98%	4%
Right Eye	69.8%	1%	96.1%	3.3%

Table 2: Eye detection results for the subset of eye annotated face, i.e. to 4059 of the total number of images.

Table 1 and 2 present the results obtained after processing the whole set of sequences with the different detectors. Observing Table 1, the Rowley's detector is notably slower than the others, but it provides eye detection in many circumstances, feature which is not considered by the Viola-Jones' detector. As for our detector, it is observed that it performs more than twice faster than the Viola-Jones' detector, and almost ten times faster than the Rowley's detector, using a PIV 2.2Ghz. 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. For our detector, false detections are in many cases associated to detections which have not been properly sized.

Some detection examples are presented in Figure 5, where each color specifies the cue used for that particularly detection. As described in that figure, some of those detections are not provided by the Viola-Jones' based detectors, but by the temporal coherence instanced in multilevel tracking and color. For those detections, their eyes were also located in 90% of them as seen in Table 1. It must be observed that eyes are located only for frontal poses in the current implementation. The eye detection error analysis described in Table 2 reflects an evident improvement in comparison to the Rowley's detector.

In at least 10 of the sequences there were detections which correspond to non face patterns (provided by the Viola-Jones detectors integrated). 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.

Figure 5: 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 recover its vertical position and fused with the latent detection thread.

Only for 3 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 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.

For multiple individuals sequences, the system needs more time as more faces are tracked simultaneously, in our experiments around 20 msecs. 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. It must also be noticed that in these sequences as no appearance cue is used to relate a detection in the next frame with a previous one, the system is not currently able to manage coherently a situation when different detection threads can overlap, i.e., there is occlusion. It is not sure that after the occlusion between two individuals, the detection threads will be properly assigned to the new detections.

4.3 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 Figures 6 and 7.

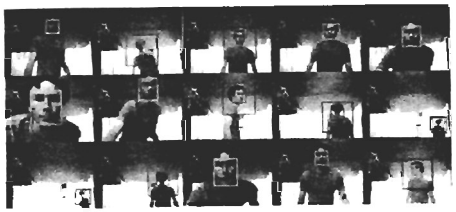

Figure 6: Sample detections corresponding to an indoor sequence (320×240 pixels).

The face location for the sequence corresponding to Figure 6 has been manually annotated. Table 3 presents the detection rates summary. For the 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. The Rowley's face detector would present the same problem.

Figure 7: Sample detections corresponding to an outdoor sequence (720×576 pixels).

On the other hand the local context detector is able to get a better detection rate. Our system, which integrates both detectors added to the temporal coherence, outperforms clearly both approaches applied to a context closer to reality.

Detector	Det rate	False det. rate
Object Centered [25]	30.5%	0.0%
Local Context [12]	66%	1.4%
Our detector	81.8%	0.3%

Table 3: Results for the indoor sequence, see Figure 6.

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. The resulting system outperforms well known face detection systems. The system is able to detect multiple faces and their eyes providing for the experiments an average processing rate of 45.6 msecs. per frame which makes the system suitable for further processing in the field of perceptual user interfaces.

Future work will focus on the improvement of the color module, and the detection of additional facial features in order to provide more elements to manage out of plane rotations.

Acknowledgments

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

References

[1] Stan Birchfield. Elliptical head tracking using intensity gradients and color histograms. In *Proc. IEEE Conf. on CVPR*, 1998.

[2] Marco La Cascia and Stan 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):322–336, April 2000.

[3] M. Castrillón Santana, F.M. Hernández Tejera, and J. Cabrera Gámez. Encara: real-time detection of frontal faces. In *International Conference on Image Processing*, Barcelona, Spain, September 2003.

[4] Stefan Feyrer and Andreas Zell. Detection, tracking and pursuit of humans with autonomous mobile robot. In *Proc. of International Conference on Intelligent Robots and Systems, Kyongju, Korea*, pages 864–869, 1999.

[5] Cayetano Guerra Artal. *Contribuciones al seguimiento visual precategórico*. PhD thesis, Universidad de Las Palmas de Gran Canaria, Octubre 2002.

[6] Erik Hjelmas and Ivar Farup. Experimental comparison of face/non-face classifiers. In *Procs. of the Third International Conference on Audio- and Video-Based Person Authentication. Lecture Notes in Computer Science 2091*, 2001.

[7] Erik Hjelmas and Boon Kee Low. Face detection: A survey. *Computer Vision and Image Understanding*, 83(3), 2001.

[8] Intel. Intel open source computer vision library, b4.0. www.intel.com/research/mrl/research/opencv, August 2004.

[9] Oliver Jesorsky, Klaus J. Kirchberg, and Robert W. Frischholz. Robust face detection using the hausdorff distance. *Lecture Notes in Computer Science. Procs. of the Third International Conference on Audio- and Video-Based Person Authentication*, 2091:90–95, 2001.

[10] Michael J. Jones and James M. Rehg. Statistical color models with application to skin detection. Technical Report Series CRL 98/11, Cambridge Research Laboratory, December 1998.

[11] Y. Kirby and L. Sirovich. Application of the karhunen-love procedure for the characterization of human faces. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 12(1), July 1990.

[12] Hannes Kruppa, Modesto Castrillón Santana, and Bernt Schiele. Fast and robust face finding via local context. In *Joint IEEE Internacional Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance (VS-PETS)*, October 2003.

[13] Stan Z. Li, Long Zhu, ZhenQiu Zhang, Andrew Blake, HongJiag Zhang, and Harry Shum. Statistical learning of multi-view face detection. In *European Conference Computer Vision*, 2002.

[14] Rainer Lienhart, Alexander Kuranov, and Vadim Pisarevsky. Empirical analysis of detection cascades of boosted classifiers for rapid object detection. In *DAGM'03*, Magdeburg, Germany, September 2003.

[15] Christine L. Lisetti and Diane 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):185–235, 2000.

[16] C. Papageorgiou, M. Oren, and T. Poggio. A general framework for object detection. In *Proceedings of the International Conference on Computer Vision*, 1998.

[17] Alex Pentland. Looking at people: Sensing for ubiquitous and wearable computing. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, January 2000.

[18] Henry A. Rowley, Shumeet Baluja, and Takeo Kanade. Neural network-based face detection. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 20(1):23–38, 1998.

[19] Henry Schneiderman and Takeo Kanade. A statistical method for 3d object detection applied to faces and cars. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2000.

[20] Karin Sobottka and Ioannis Pitas. A novel method for automatic face segmentation, face feature extraction and tracking. *Signal Processing: Image Communication*, 12(3), 1998.

[21] Moritz Storring, Hans J. Andersen, and Erik Granum. Physics-based modelling of human skin colour under mixed illuminants. *Robotics and Autonomous Systems*, 2001.

[22] Kah-Kay Sung and Tomasso Poggio. Example-based learning for view-based human face detection. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 20(1), January 1998.

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

[24] V. Vapnik. *The nature of statistical learning theory*. Springer, New York, 1995.

[25] Paul Viola and Michael J. Jones. Rapid object detection using a boosted cascade of simple features. In *Computer Vision and Pattern Recognition*, 2001.

[26] Paul Viola, Michael J. Jones, and Daniel Snow. Detecting pedestrians using patterns of motion and appearance. In *Proc. of the International Conference on Computer Vision*, October 2003.

[27] Christopher Wren, Ali Azarbayejani, Trevor Darrell, and Alex Pentland. Pfinder: Real-time tracking of the human body. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 19(7), July 1997.

[28] Ming-Hsuan Yang, David Kriegman, and Narendra Ahuja. Detecting faces in images: A survey. *Transactions on Pattern Analysis and Machine Intelligence*, 24(1):34–58, 2002.