Edge projections for eye localization

Mehmet Turkan
Bilkent University
Department of Electrical and Electronics Engineering
Bilkent, 06800 Ankara, Turkey
E-mail: turkan@ee.bilkent.edu.tr

Montse Pardas
Technical University of Catalonia
Department of Signal Theory and Communications
08034 Barcelona, Spain

A. Enis Cetin
Bilkent University
Department of Electrical and Electronic Engineering
Bilkent, 06800 Ankara, Turkey

Abstract. An algorithm for human-eye localization in images is presented for faces with frontal pose and upright orientation. A given face region is filtered by a highpass wavelet-transform filter. In this way, edges of the region are highlighted, and a caricature-like representation is obtained. Candidate points for each eye are detected after analyzing horizontal projections and profiles of edge regions in the highpass-filtered image. All the candidate points are then classified using a support vector machine. Locations of each eye are estimated according to the most probable ones among the candidate points. It is experimentally observed that our eye localization method provides promising results for image-processing applications. *© 2008 Society of Photo-Optical Instrumentation Engineers.* [DOI: 10.1117/1.2902437]

Subject terms: eye localization; face detection; wavelet transform; edge projections; support-vector machines.

Paper 070797R received Sep. 23, 2007; revised manuscript received Jan. 25, 2008; accepted for publication Jan. 28, 2008; published online Apr. 29, 2008.

1 Introduction

The problem of human-eye detection, localization, and tracking has received significant attention during the past several years because of the wide range of human-computer interaction (HCI) and surveillance applications. Because eyes are one of the most important salient features of a human face, detecting and localizing them helps researchers working on face detection, face recognition, iris recognition, facial expression analysis, etc.

In recent years, many heuristic and pattern-recognition-based methods have been proposed to detect and localize eyes in still images and video. Most of these methods described in the literature, ranging from very simple algorithms to composite high-level approaches, are closely associated with face detection and face recognition. Traditional image-based eye detection methods assume that the eyes appear different from the rest of the face in both shape and intensity. Dark pupil, white sclera, circular iris, eye corners, eye shape, etc., are specific properties of an eye to distinguish it from other objects.[1]

Morimoto and Mimica[2] reviewed the state of the art of eye-gaze trackers, comparing the strengths and weaknesses of the alternatives available today. They also improved the usability of several remote eye-gaze tracking techniques. Zhou and Geng[3] developed a method for detecting eyes with projection functions. After localizing the rough eye positions using Wu and Zhou's[4] method, they expand a rectangular area near each rough position. Special cases of the generalized projection function (GPF), viz., the integral projection function (IPF) and the variance projection function (VPF), are used to localize the central positions of eyes in eye windows. The IPF and VPF are calculated from the pixel intensity values both horizontally and vertically.

Recently, wavelet-domain[5,6] feature extraction methods have been developed and become very popular for face and eye detection.[7,8] Zhu et al.[8] described a subspace approach to capture local discriminative features in the space-frequency domain for fast face detection based on orthonormal wavelet packet analysis. They demonstrated that the detail (high-frequency subband) information within local facial areas contains information about the eyes, nose, and mouth, and offers notable discrimination ability for face detection. This fact may also be used to detect and localize facial areas such as eyes.

Cristinacce et al.[9] developed a multistage approach to detect and locate 17 feature points on a human face, including the eyes. After applying a face detector[10] to find the approximate scale and location of the face in the image, they extract and combine individual features using a pairwise reinforcement of feature responses (PRFR) algorithm using pairwise probabilistic constraints. The estimated features are then refined using a version of the active appearance model (AMM) search, which is based on edge and corner features. In this AMM approach, normalized gradients in horizontal and vertical directions, a measure of *edgeness*, and a measure of *cornerness* are computed for each pixel.

Jesorsky et al.[11] presented a model-based face detection system, for grayscale still images, using edge features and the modified Hausdorff distance. After detecting the rough position of the facial region, face position parameters, including those for eyes, are refined as a second step. They additionally applied a multilayer perceptron (MLP) neural network, which is trained with pupil-centered images whenever the refinement results are not satisfactory. The performance of their face detection system is validated by a relative-error measure based on a comparison between the expected and the estimated eye positions.

Asteriadis et al.[12] developed an eye detection algorithm based on only the geometrical information on the eye and its surrounding area. After applying a face detector in order

0091-3286/2008/$25.00 © 2008 SPIE

to obtain the location of the face in the image, they extracted an edge map of the face region using the Canny edge detector. Then, they assigned a vector to every pixel, pointing to the closest edge pixel, containing the length (magnitude) and slope (angle) information. Eye detection and localization are finally accomplished using the eigenvector spaces obtained in principal-component analysis (PCA) of length and angle maps.

In this study, a method of human-eye localization in images is proposed with the assumption that a human face region in a given still image is already detected by means of a face detector. This method is based on the idea that eyes can be detected and localized from edges of a typical human face. In fact, a caricaturist draws a face image in a few strokes by drawing the major edges (eyes, nose, mouth, etc.), of the face. Most wavelet-domain image classification methods are also based on this fact, in that significant wavelet coefficients are closely related to edges.[5,7,13]

The proposed algorithm works with edge projections of given face images. After an approximate horizontal level detection, each eye is first localized horizontally using horizontal projections of associated edge regions. Then, horizontal edge profiles are calculated on the estimated horizontal levels. Eye candidate points are determined by pairing the local maximum point locations in the horizontal profiles with the associated horizontal levels. After obtaining the eye candidate points, verification is carried out by a classifier based on a support vector machine (SVM). The locations of the eyes are finally estimated according to the most probable point for each eye separately.

This paper is organized as follows. Section 2 describes our eye localization algorithm; each step is briefly explained with attention to the techniques used in its implementation. In Sec. 3, experimental results of the proposed algorithm are presented, and its detection performance is compared with that of currently available eye localization methods. Conclusions are given in Sec. 4.

2 Eye Localization System

In this paper, a human eye localization scheme for faces with frontal pose and upright orientation is developed. After detecting a human face in a given color image using the edge projection method proposed by Turkan et al.,[14] the face region is decomposed into its wavelet-domain subimages. The detail information within local facial areas (e.g., eyes, nose, and mouth), is obtained in low-high, high-low, and high-high subimages of the face pattern. A brief review of the face detection algorithm is given in Sec. 2.1, and the wavelet-domain processing is presented in Sec. 2.2. After analyzing horizontal projections and profiles of horizontal-crop and vertical-crop edge images, the candidate points for each eye are detected as explained in Sec. 2.3. All the candidate points are then classified using an SVM-based classifier. Finally, the location of each eye is estimated according to the most probable one among the candidate points.

2.1 *Face Detection Algorithm*

The face detection algorithm starts with skin color detection and segmentation. In this study, the illumination effect is prevented using tint-saturation-luminance (TSL) color space by discarding the luminance component. A normalized chrominance-luminance TSL space is a transformation of the normalized red-green-blue (RGB) into more intuitive values.[15] TSL color-space components can be obtained using RGB color-space values as follows:

Fig. 1 Example skin color image samples for training.

$$S = \left[\frac{9}{5}(r'^2 + g'^2)\right]^{1/2},$$

$$T = \begin{cases} \arctan(r'/g')/2\pi + 1/4, & g' > 0, \\ \arctan(r'/g')/2\pi + 3/4, & g' < 0, \\ 0, & g = 0, \end{cases}$$

$$L = 0.299R + 0.587G + 0.114B, \tag{1}$$

where $r' = r - 1/3$, $g' = g - 1/3$, and r, g are normalized components of the RGB color space.

The distribution of skin color pixels is obtained from skin color training samples, and they are represented by an elliptical Gaussian joint probability density function using the normalized tint and saturation. An example set of images used in training is shown in Fig. 1. Given a color image, each pixel is labeled as skin or nonskin according to the estimated Gaussian model. Then morphological operations are performed on skin-labeled pixels in order to have connected face candidate regions.

After determining all possible face candidate regions in a given color image, a single-stage 2-D rectangular wavelet transform of each region's intensity (grayscale) images is computed. In this way, wavelet-domain subimages are obtained. The low-high and high-low subimages contain horizontal and vertical edges of the region, respectively. The high-high subimage may contain almost all the edges, if the face candidate region is sharp enough. It is clear that the detail information within local facial areas (e.g., edges due to eyes, nose, and mouth), shows noticeable discrimination ability for detection of frontal-view faces. Turkan et al.[14] take advantage of this fact by characterizing these subimages using their projections and obtain 1-D projection feature vectors corresponding to edge images of face or face-like regions. The horizontal projection $H[\cdot]$ and vertical projection $V[\cdot]$ are simply computed by summing normalized pixel values $d[\cdot,\cdot]$ in a row and column, respectively:

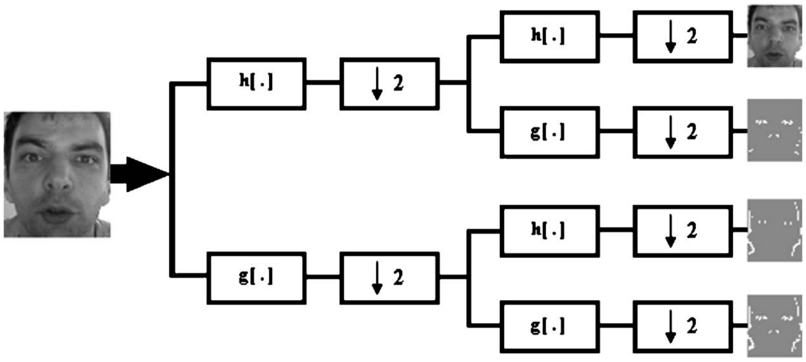

Fig. 2 Two-dimensional rectangular wavelet decomposition of a face pattern: low-low, low-high, high-low, and high-high subimages. Here $h[\cdot]$ and $g[\cdot]$ represent 1-D lowpass and highpass filters, respectively.

$$H[y] = \frac{1}{m}\sum_{x} |d[x,y]|,$$

$$V[x] = \frac{1}{k}\sum_{y} |d[x,y]|, \tag{2}$$

where $d[x,y]$ is the sum of the absolute values of the three high-band subimages, and k and m are the numbers of rows and columns, respectively.

Furthermore, Haar filter-like projections are computed, as in Viola and Jones's[10] approach, as additional feature vectors that are obtained from differences of two subregions in the candidate region. The final feature vector for a face candidate region is obtained by concatenating all the horizontal, vertical, and filterlike projections. These feature vectors are then classified using an SVM-based classifier into face and nonface classes.

Because wavelet-domain processing is used both for face and eye detection, it is described in more detail in the next subsection.

2.2 *Wavelet Decomposition of Face Patterns*

A given face region is processed using a 2-D filter bank. The region is first processed rowwise using a 1-D Lagrange filter bank[16] with a lowpass-highpass filter pair, $h[n] = \{0.25, 0.5, 0.25\}$ and $g[n] = \{-0.25, 0.5, -0.25\}$, respectively. The resulting two image signals are then processed columnwise, using the same filter bank. The high-band subimages that are obtained using a highpass filter contain edge information; e.g., the low-high and high-low subimages contain horizontal and vertical edges of the input image, respectively (see Fig. 2). The absolute values of low-high, high-low, and high-high subimages can be summed to obtain an image having the significant edges of the face region.

A second approach is to use a 2-D lowpass filter and subtract the lowpass-filtered image from the original image. The resulting image also contains the edge information of the original image, and it is equivalent to the sum of undecimated low-high, high-low, and high-high subimages, which we call the detail image, as shown in Fig. 3(b).

2.3 *Feature Extraction and Eye Localization*

The first step of feature extraction is denoising. The detail image of a given face region is denoised by soft thresholding, using the method of Donoho and Johnstone.[17] The threshold value t_n is obtained as follows:

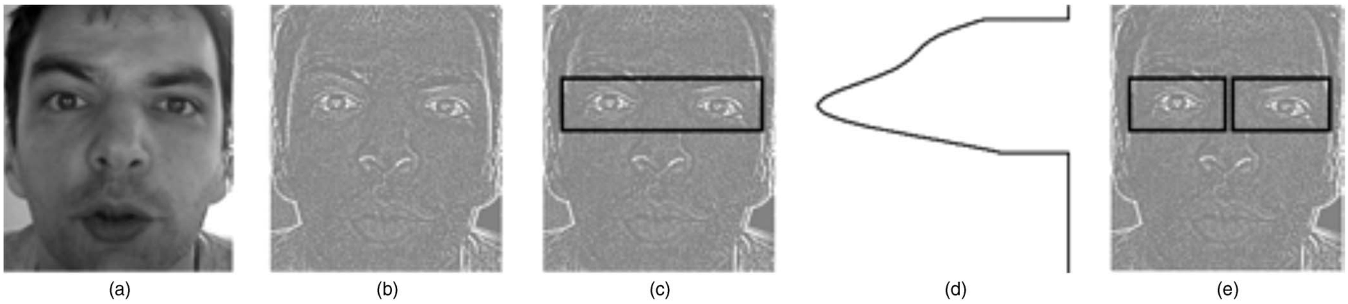

Fig. 3 (a) An example face region with its (b) detail image and (c) horizontal-crop edge image covering the eyes region, determined according to (d) smoothed horizontal projection in the upper part of the detail image (the projection data is filtered with a narrowband lowpass filter to obtain the smooth projection plot). Vertical-crop edge regions are obtained by cropping the horizontal-crop edge image vertically as shown in (e).

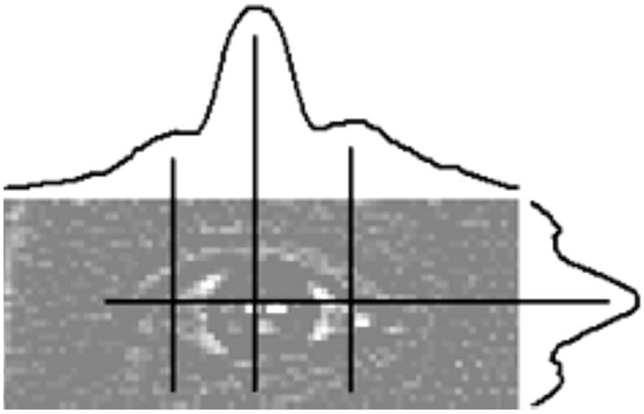

Fig. 4 An example vertical-crop edge region with its smoothed horizontal projection and profile. Eye candidate points are obtained by pairing the maxima in the horizontal profile with the associated horizontal level.

$$t_n = \left(\frac{2 \log n}{n} \right)^{1/2} \sigma, \tag{3}$$

where n is the number of wavelet coefficients in the region and σ is the estimated standard deviation of the Gaussian noise in the input signal. The wavelet coefficients below the threshold are changed to zero, and those above the threshold are kept as they are. This initial step removes the noise effectively while preserving the edges in the data.

The second step of the algorithm is to determine the approximate horizontal position of the eyes, using the horizontal projection in the upper part of the detail image, because the eyes are located in the upper part of a typical human face [see Fig. 3(a)]. This provides robustness against the effects of edges due to neck, mouth (teeth), and nose on the horizontal projection. The index of the global maximum in the smoothed horizontal projection in this region indicates the approximate horizontal location of both eyes, as shown in Fig. 3(d). On obtaining a rough horizontal position, the detail image is cropped horizontally according to the maximum as shown in Fig. 3(c). Then, vertical-crop edge regions are obtained by cropping the horizontally cropped edge image into two parts vertically, as shown in Fig. 3(e).

The third step is to compute again horizontal projections in both right-eye and left-eye vertical-crop edge regions in order to detect the exact horizontal positions of each eye separately. The global maximum of these horizontal projections for each eye provides the estimated horizontal levels. This approach of dividing the image into two vertical-crop regions provides some freedom in detecting eyes in oriented face regions where the eyes are not located on the same horizontal level.

Since a typical human eye consists of a white sclera around a dark pupil, the transition from white to dark (or dark to white) produces significant jumps in the coefficients of the detail image. We take advantage of this fact by calculating horizontal profiles on the estimated horizontal levels for each eye. The jump locations are estimated from smoothed horizontal profile curves. An example vertical-crop edge region with its smoothed horizontal projection and profile is shown in Fig. 4. It is worth mentioning that the global maximum in the smoothed horizontal profile signals is due to the transitions both from white sclera to pupil and from pupil to white sclera. The first and last peaks correspond to outer and inner eye corners. Since there is a transition from skin to white sclera (or white sclera to skin), these peak values are small compared to those for the transition from white sclera to pupil (or pupil to white sclera). However, this may not be the case in some eye regions. There may be more (or less) than three peaks, depending on the sharpness of the vertical-crop eye region and eyeglasses.

Eye candidate points are obtained by pairing the indices of the maxima in the smoothed horizontal profiles with the associated horizontal levels for each eye. An example horizontal level estimate with its candidate vertical positions is shown in Fig. 4.

An SVM-based classifier is used to discriminate the possible eye candidate locations. A rectangle centered on each candidate point is automatically cropped and fed to the

Table 1 Eye localization results.

		Success rate (%)	
Method	Database	$d<0.25$	$d<0.10$
Our method (edge projections)	CVL	99.70	80.90
Our method (edge projections)	BioID	99.46	73.68
Asteriadis et al. (2006)	BioID	97.40	81.70
Zhou and Geng (2004)	BioID	94.81	—
Jesorsky et al. (1992)	BioID	91.80	79.00
Cristinacce et al. (2004)	BioID	98.00	96.00

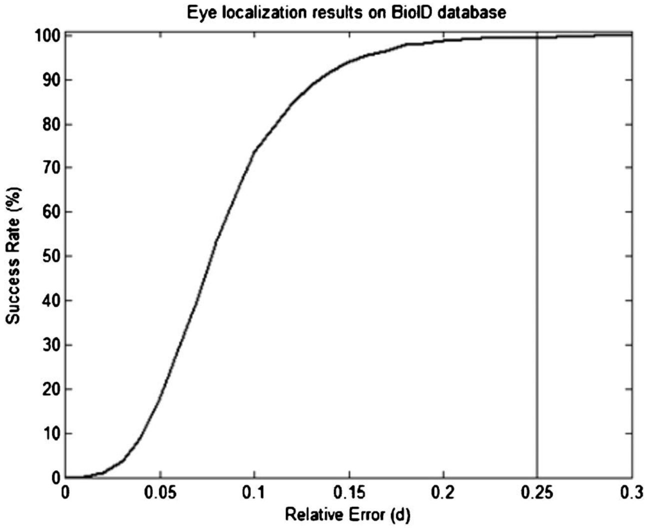

Fig. 5 Distribution function of relative eye distances given by our algorithm on the BioID database.

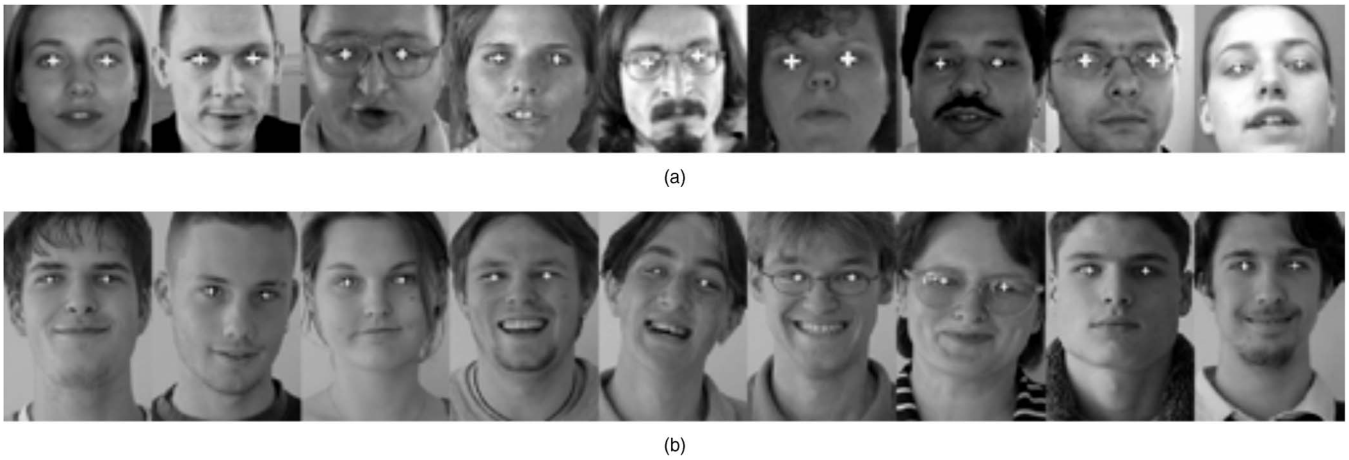

Fig. 6 Examples of estimated eye locations from the (a) BioID, (b) CVL database.

SVM classifier. The size of the rectangles depends on the resolution of the detail image. However, the cropped rectangular region is finally resized (using bicubic interpolation) to a resolution of 25 × 25 pixels. The feature vectors for each eye candidate region are calculated similarly to the face detection algorithm by concatenating the horizontal and vertical projections of the rectangles around eye candidate locations. The points that are classified as an eye by the SVM classifier are then ranked with respect to their estimated probability values,[18] produced also by the classifier. The locations of eyes are finally determined according to the most probable point for each eye separately.

In this paper, we used a library for SVMs called LIBSVM.[19] Our simulations are carried out in a C++ environment with interface for Python using a radial basis function (RBF) as the kernel. The LIBSVM package provides the necessary quadratic programming routines to carry out the classification. It performs cross validation on the feature set and also normalizes each feature by linearly scaling it to the range $[-1,+1]$. This package also contains a multiclass probability estimation algorithm proposed by Wu et al.[18]

3 Experimental Results

The proposed eye localization algorithm is evaluated in this paper on the CVL (http://www.lrv.fri.uni-lj.si/) and BioID (http://www.bioid.com/) face databases. All the images in these databases are of head and shoulders.

The CVL database contains 797 color images of 114 persons. Each person has seven different images of size 640 × 480 pixels: a far-left side view, 45-deg-angle side view, serious-expression frontal view, 135-deg-angle side view, far-right side view, smile (showing no teeth) frontal view, and smile (showing teeth) frontal view. Since the developed algorithm can only be applied to faces with frontal pose and upright orientation, our experimental data set contains 335 frontal-view face images from this database. Face detection is carried out using Ref. 14 for this data set, since the images are colored.

The BioID database consists of 1521 gray-level images of 23 persons with a resolution of 384 × 286 pixels. All images in this database are frontal views of faces with a large variety of illumination conditions and face sizes. Face detection is carried out using Intel's OpenCV face detection method (http://www.intel.com/), since all images are gray level.

The estimated eye locations are compared with the exact eye center locations based on a relative error measure proposed in Ref. 11. Let C_r and C_l be the exact eye center locations, and $\tilde{C}_r$ and $\tilde{C}_l$ be the estimated eye positions. The relative error of this estimation is measured according to the formula

$$d = \frac{\max(\|C_r - \tilde{C}_r\|, \|C_l - \tilde{C}_l\|)}{\|C_r - C_l\|}. \tag{4}$$

In a typical human face, the width of a single eye roughly equals the distance between inner eye corners. Therefore, half an eye width approximately equals a relative error of 0.25. Thus, in this paper we consider a relative error of $d < 0.25$ to indicate a correct estimation of eye positions.

Our method has a 99.46% overall success rate for $d < 0.25$ on the BioID database, whereas Jesorsky et al.[11] achieved 91.80%, and Zhou and Geng[3] 94.81%. Asteriadis et al.[12] reported a success rate 97.40% using the same face detector on this database. Cristinacce et al.[9] had a success rate 98.00% (we obtained this value from their distribution function given by the relative-eye-distance graph). However, our method reaches 73.68% for $d < 0.10$, whereas Ref. 11 had 79.00%, Ref. 12 81.70%, and Ref. 9 96.00% for this strict d value. All the experimental results are given in Table 1, and the distribution function of relative eye distances in the BioID database is shown in Fig. 5. Some examples of estimated eye locations are shown in Fig. 6.

4 Conclusion

In this paper, we have presented a human-eye localization algorithm for faces with frontal pose and upright orientation. The performance of the algorithm has been examined on two face databases by comparing the estimated eye positions with the ground-truth values using a relative-error measure. The localization results show that our algorithm is robust against both illumination and scale changes, since the BioID database contains images with a large variety of illumination conditions and face size. To the best of our knowledge, our algorithm gives the best results on the BioID database for $d < 0.25$. Therefore, it can be used for HCI applications, and as the initialization stage of eye-trackers. In eye tracking applications (e.g., Ref. 20), a good initial estimate is satisfactory, for the tracker further localizes the positions of eyes. For this reason, the $d < 0.25$ results are more important than those of $d < 0.10$ from the tracker point of view.

Our algorithm provides approximately 1.5% improvement over the other methods. This may not look so good at first glance, but it is a significant improvement for commercial application, for it corresponds to one more satisfied customer in a group of hundred users.

Acknowledgment

This work is supported by the European Commission Network of Excellence FP6-507752 Multimedia Understanding through Semantics, Computation and Learning (MUSCLE-NoE [http://www.muscle-noe.org/]), and FP6-511568 Integrated Three-Dimensional Television—Capture, Transmission, and Display (3DTV-NoE [https://www.3dtv-research.org/]) projects.

References

1. Z. Zhu and Q. Ji, "Robust real-time eye detection and tracking under variable lighting conditions and various face orientations," *Comput. Vis. Image Underst.* **98**, 124–154 (2005).
2. C. H. Morimoto and M. R. M. Mimica, "Eye gaze tracking techniques for interactive applications," *Comput. Vis. Image Underst.* **98**, 4–24 (2005).
3. Z. H. Zhou and X. Geng, "Projection functions for eye detection," *Pattern Recogn.* **37**(5), 1049–1056 (2004).
4. J. Wu and Z. H. Zhou, "Efficient face candidates selector for face detection," *Pattern Recogn.* **36**(5), 1175–1186 (2003).
5. S. G. Mallat, "A theory for multiresolution signal decomposition, the wavelet representation," *IEEE Trans. Pattern Anal. Mach. Intell.* **11**, 674–693 (1989).
6. I. Daubechies, "The wavelet transform, time-frequency localization and signal analysis," *IEEE Trans. Inf. Theory* **36**, 961–1005 (1990).
7. C. Garcia and G. Tziritas, "Face detection using quantized skin color regions merging and wavelet packet analysis," *IEEE Trans. Multimedia* **1**, 264–277 (1999).
8. Y. Zhu, S. Schwartz, and M. Orchard, "Fast face detection using subspace discriminant wavelet features," in *Proc. Conf. on Computer Vision and Pattern Recognition*, Vol. **I**, pp. 636–642, IEEE (2000).
9. D. Cristinacce, T. Cootes, and I. Scott, "A multistage approach to facial feature detection," in *Proc. Br. Machine Vision Conf.*, pp. 231–240 (2004).
10. P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Proc. Computer Soc. Conf. on Computer Vision and Pattern Recognition*, Vol. **I**, pp. 511–518, IEEE (2001).
11. O. Jesorsky, K. J. Kirchberg, and R. W. Frischholz, "Robust face detection using the Hausdorff distance," In *Proc. Int. Conf. on Audio- and Video-Based Biometric Person Authentication*, pp. 90–95, Springer (1992).
12. S. Asteriadis, N. Nikolaidis, A. Hajdu, and I. Pitas, "An eye detection algorithm using pixel to edge information," in *Proc. Int. Symp. on Control, Communications, and Signal Processing*, IEEE (2006).
13. A. E. Cetin and R. Ansari, "Signal recovery from wavelet transform maxima," *IEEE Trans. Signal Process.* **42**, 194–196 (1994).
14. M. Turkan, B. Dulek, I. Onaran, and A. E. Cetin, "Human face detection in video using edge projections," in *Visual Information Processing XV*, *Proc. SPIE* **6246**, 624607 (2006).
15. V. Vezhnevets, V. Sazonov, and A. Andreeva, "A survey on pixel-based skin color detection techniques," in *Proc. Graphicon'03* (2003).
16. C. W. Kim, R. Ansari, and A. E. Cetin, "A class of linear-phase regular biorthogonal wavelets," in *Proc. Int. Conf. on Acoustics, Speech, and Signal Processing*, Vol. **IV**, pp. 673–676, IEEE (1992).
17. D. L. Donoho and I. M. Johnstone, "Ideal spatial adaptation via wavelet shrinkage," *Biometrika* **81**, 425–455 (1994).
18. T. F. Wu, C. J. Lin, and R. C. Weng, "Probability estimates for multi-class classification by pairwise coupling," *J. Mach. Learn. Res.* **5**, 975–1005 (2004).
19. C. C. Chang and C. J. Lin, *LIBSVM: a library for support vector machines*, software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm (2001).
20. A. M. Bagci, R. Ansari, A. Khokhar, and A. E. Cetin, "Eye tracking using Markov models," in *Proc. Int. Conf. on Pattern Recognition*, Vol. **III**, pp. 818–821, IEEE (2004).

Mehmet Turkan received his MS degree in 2006 in electrical and electronics engineering from Bilkent University, Ankara, Turkey, where he is currently a PhD student and a research and teaching assistant. He received his BS degree from the Electrical and Electronics Engineering Department of Osmangazi University, Eskisehir, Turkey, in 2004. His research interests are in the area of signal processing with an emphasis on image and video processing, pattern recognition and classification, and computer vision.

Montse Pardas received the MS degree in telecommunications and the PhD degree from the Polytechnic University of Catalonia, Barcelona, Spain, in July 1991 and January 1995, respectively. Since September 1994 she has been teaching communication systems and digital image processing in that university, where she is currently an associate professor. From January 1999 to December 1999 she was a research visitor at Bell Labs, Lucent Technologies, in New Jersey. Her main research activity deals with image and sequence analysis, with a special emphasis on segmentation, motion estimation, tracking, and face analysis. She has participated in the research activities of several Spanish and European projects. She was an associate editor of the *Eurasip Journal of Applied Signal Processing* from 2004 to 2006, and belongs to the technical committee of several image-processing conferences and workshops.

A. Enis Cetin studied electrical engineering at the METU, Ankara, Turkey. After getting his BS degree, he got his MSE and PhD degrees in systems engineering from the Moore School of Electrical Engineering at the University of Pennsylvania in Philadelphia, USA. Between 1987 and 1989, he was an assistant professor of electrical engineering at the University of Toronto, Canada. Since then he has been with Bilkent University, Ankara, Turkey. In 1996, he was promoted to the position of professor. During summers of 1988, 1991, 1992 he was with Bell Communications Research (Bellcore) as a consultant, and resident visitor. He spent the 1996–1997 academic year at the University of Minnesota, Minneapolis, USA, as a visiting professor. He was a member of the DSP technical committee of the IEEE Circuits and Systems Society. He founded the Turkish Chapter of the IEEE Signal Processing Society in 1991. He is a senior member of IEEE and EURASIP. He was the co-chair of the IEEE-EURASIP Nonlinear Signal and Image Processing Workshop (NSIP'99), which was held in 1999 in Antalya, Turkey. He organized a special session on education at NSIP-99. He was the technical co-chair of the European Signal Processing Conference (EUSIPCO-2005). He received the Young Scientist Award of TUBITAK (the Turkish Scientific and Technical Research Council) in 1993. Between 1999 and 2003, Prof. Cetin was an associate editor of the *IEEE Transactions on Image Processing*. Currently, he is on the editorial boards of *EURASIP Journal of Advances in Signal Processing* and *Signal Processing*.