Combining Head Pose and Eye Location Information for Gaze Estimation

Roberto Valenti, *Member, IEEE*, Nicu Sebe, *Member, IEEE*, and Theo Gevers, *Member, IEEE*

***Abstract*—Head pose and eye location for gaze estimation have been separately studied in numerous works in the literature. Previous research shows that satisfactory accuracy in head pose and eye location estimation can be achieved in constrained settings. However, in the presence of nonfrontal faces, eye locators are not adequate to accurately locate the center of the eyes. On the other hand, head pose estimation techniques are able to deal with these conditions; hence, they may be suited to enhance the accuracy of eye localization. Therefore, in this paper, a hybrid scheme is proposed to combine head pose and eye location information to obtain enhanced gaze estimation. To this end, the transformation matrix obtained from the head pose is used to normalize the eye regions, and in turn, the transformation matrix generated by the found eye location is used to correct the pose estimation procedure. The scheme is designed to enhance the accuracy of eye location estimations, particularly in low-resolution videos, to extend the operative range of the eye locators, and to improve the accuracy of the head pose tracker. These enhanced estimations are then combined to obtain a novel visual gaze estimation system, which uses both eye location and head information to refine the gaze estimates. From the experimental results, it can be derived that the proposed unified scheme improves the accuracy of eye estimations by 16% to 23%. Furthermore, it considerably extends its operating range by more than 15° by overcoming the problems introduced by extreme head poses. Moreover, the accuracy of the head pose tracker is improved by 12% to 24%. Finally, the experimentation on the proposed combined gaze estimation system shows that it is accurate (with a mean error between 2° and 5°) and that it can be used in cases where classic approaches would fail without imposing restraints on the position of the head.**

***Index Terms*—Eye center location, gaze estimation, head pose estimation.**

I. Motivation and Related Work

IMAGE-BASED gaze estimation is important in many applications, spanning from human–computer interaction (HCI) to human behavior analysis. In applications where human activity is under observation from a static camera, the estimation of the visual gaze provides important information about the interest of the subject, which is commonly used as control devices for disabled people [1], i.e., to analyze the user attention while driving [14], and other applications. It is known that the gaze is a product of two contributing factors [26], i.e., the head pose and the eye locations. The estimation of these two factors is often achieved using expensive, bulky, or limiting hardware [10]. Therefore, the problem is often simplified by either considering the head pose or the eye center locations as the only feature to understand the interest of a subject [27], [36].

There is an abundance of literature concerning these two topics separately; recent surveys on eye center location and head pose estimation can be found in [17] and [31]. The *eye location* algorithms found in commercially available eye trackers share the problem of sensitivity to head pose variations and require the user to be either equipped with a head-mounted device or to use a high-resolution camera combined with a chin rest to limit the allowed head movement. Furthermore, daylight applications are precluded due to the use of active infrared (IR) illumination to obtain accurate eye location through corneal reflection. The appearance-based methods that make use of standard low-resolution cameras are considered to be less invasive and, thus, more desirable in a large range of applications. Within the appearance-based methods for eye location proposed in the literature [22], [24], [33], [44], reported results support the evidence that accurate appearance-based eye center localization is becoming feasible and that it could be used as an enabling technology for a various set of applications.

Head pose estimation often requires multiple cameras, or complex face models, which requires accurate initialization. Ba and Odobez [4] improve the accuracy of pose estimates and of the head tracking by considering these as two coupled problems in a probabilistic setting within a mixed-state particle filter framework. They refine this method by the fusion of four camera views in [5]. Huang and Trivedi propose to integrate a skin-tone edge-based detector into a Kalman-filter-based robust head tracker and hidden-Markov-model-based pose estimator in [19]. Hu *et al.* describe a coarse-to-fine pose estimation method by combining facial appearance asymmetry and 3-D head model [18]. A generic 3-D face model and an ellipsoidal head model are utilized in [2] and [41], respectively. In [30], an online tracking algorithm employing adaptive view-based appearance models is proposed. The method provides drift-free tracking by maintaining a dynamic set of keyframes with views of the head under various poses and registering the current frame to the previous frames and keyframes.

Although several head pose or eye location methods have shown success in gaze estimation, the underlying assumption

Manuscript received June 27, 2010; revised May 17, 2011; accepted July 01, 2011. Date of publication July 22, 2011; date of current version January 18, 2012. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Miles N. Wernick.

R. Valenti is with the Intelligent Systems Lab, Amsterdam, University of Amsterdam, 1098 XH Amsterdam, The Netherlands (e-mail: r.valenti@uva.nl).

N. Sebe is with the Department of Information Engineering and Computer Science, University of Trento, 38123 Trento, Italy (e-mail: nicu.sebe@disi.unitn.it).

T. Gevers is with the Intelligent Systems Lab, Amsterdam, University of Amsterdam, 1098 XH Amsterdam, The Netherlands, and also with the Computer Vision Center, Universitat Autònoma de Barcelona, 08193 Barcelona, Spain (e-mail: th.gevers@uva.nl).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIP.2011.2162740

of being able to estimate the gaze starting from eye location or head pose only is valid in a limited number of scenarios [38], [48]. For instance, if we consider an environment composed of a target scene (a specific scene under analysis, such as a computer monitor, an advertising poster, a shelf, etc.) and a monitored area (the place from which the user looks at the target scene), an eye gaze tracker alone would fail when trying to understand which product on the shelf is being observed, whereas an head pose gaze estimator alone would fail in finely controlling the cursor on a computer screen.

Hence, a number of studies focused on *combining head* and *eye information* for gaze estimation are available in the literature. Newman and Matsumoto [28] and Matsumoto *et al.* [32] consider a tracking scenario equipped with stereo cameras and employ 2-D feature tracking and 3-D model fitting. The work proposed by in [21] describe a real-time eye, gaze, and head pose tracker for monitoring driver vigilance. The authors use IR illumination to detect the pupils and derive the head pose by building a feature space from them. Although their compound tracking property promote them against separate methods, the practical limitations and the need for improved accuracy make them less attractive in comparison with monocular low-resolution implementations.

However, no study is performed on the feasibility of an accurate appearance-only gaze estimator that considers both the head pose and eye location factors. Therefore, our goal is to develop a system capable of analyzing the visual gaze of a person starting from monocular video images. This allows studying the movement of the user's head and eyes in a more natural manner than traditional methods, as there are no additional requirements needed to use the system.

To this end, we propose a unified framework for head pose and eye location estimation for visual gaze estimation. The head tracker is initialized using the location and the orientation of the eyes, whereas the latter ones are obtained by pose-normalized eye patches obtained from the head tracker. A feedback mechanism is employed in the evaluation of the tracking quality. When the two modules do not yield concurring results, both are adjusted to get in line with each other, aiming to improve the accuracy of both tracking schemes. The improved head pose estimation is then used to define the field of view, while displacement vectors between the pose-normalized eye locations and their resting positions are used to adjust the gaze estimation obtained by the head pose only. In this way, a novel multimodal visual gaze estimator is obtained.

The contributions of this paper are the following:

1) Rather than just a sequential combination, we propose a unified framework that provides a deep integration of the used head pose tracker and eye location estimation methods.
2) The normal working range of the used eye locator ($\sim 30^\circ$) is extended. The shortcomings of the reported eye locators due to extreme head poses are compensated using the feedback from the head tracker.
3) Steered by the obtained eye location, the head tracker provides better pose accuracy and can better recover the correct pose when the head tracker is lost.
4) The eye location and head pose information are used together in a multimodal visual gaze estimation system, which uses the eyes to adjust the gaze location determined by the head pose.

This paper is structured as follows: The reason behind the choice and the theory of the used eye locator and head pose estimator will be discussed in Section II. In Section III, the discussed components will be combined in a synergetic way so that the eye locator will be aided by the head pose and the head pose estimator will be aided by the obtained eye locations. Section IV will describe how the improved estimations could be used to create a combined gaze estimation system. In Section V, three independent experiments will analyze the improvements obtained on the head pose, eye location, and combined gaze estimation. Finally, the discussions and the conclusions will be given in Section VI.

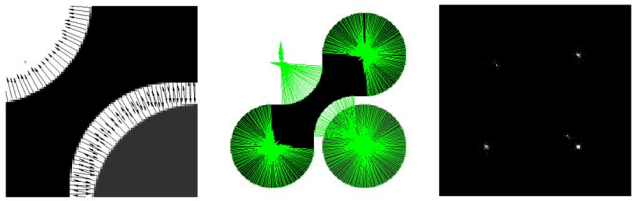
II. Eye Location and Head Pose Estimation

To describe how the used eye locator and head pose estimator are combined in Section III, here, the used eye locator and head pose estimator are discussed.

A. Eye Center Localization

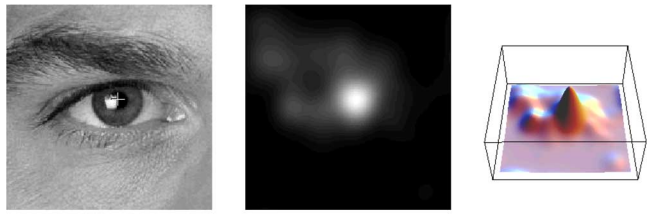
As we are discussing appearance-based methods here, an overview of the state of the art on the subject is given. The method used in [3] assigns a vector to every pixel in the edge map of the eye area, which points to the closest edge pixel. The length and the slope information of these vectors is consequently used to detect and localize the eyes by matching them with a training set. Cristinacce *et al.* [12] use a multistage approach to detect facial features (among them are the eye centers) using a face detector, pairwise reinforcement of feature responses, and a final refinement by using an active appearance model (AAM) [11]. Türkan *et al.* [42] use edge projection (GPF) [51] and support vector machines (SVMs) to classify estimates of eye centers. Bai *et al.* [6] use an enhanced version of Reisfeld's *generalized symmetry transform* [35] for the task of eye location. Hamouz *et al.* [16] search for ten features using Gabor filters, use feature triplets to generate face hypothesis, register them for affine transformations, and verify the remaining configurations using two SVM classifiers. Finally, Campadelli *et al.* [8] use an eye detector to validate the presence of a face and to initialize an eye locator, which, in turn, refines the position of the eye using the SVM on optimally selected Haar wavelet coefficients. With respect to the aforementioned methods, the method proposed in [44] achieves the best results for accurate eye center localization, without heavy constraints on illumination, rotation, and robust to slight pose changes, and will be therefore used in this paper.

The method uses isophote (i.e., curves connecting points of equal intensity) properties to obtain the center of (semi) circular patterns. This idea is based on the observation that the eyes are characterized by radially symmetric brightness patterns; hence, it looks for the center of the curved isophotes in the image. In Cartesian coordinates, the isophote curvature k is expressed as



(a) (b) (c)

Fig. 1. (a) Direction of the gradient under the image's edges, (b) the displacement vectors pointing to the isophote centers, and (c) the centermap.



(a) (b) (c)

Fig. 2. (a) Source image, (b) the obtained centermap, and (c) the 3-D representation of the latter.

$$\kappa = -\frac{\frac{\delta I}{\delta y}^2 \frac{\delta^2 I}{\delta x^2} - 2\frac{\delta I}{\delta x}\frac{\delta^2 I}{\delta x \delta y}\frac{\delta I}{\delta y} + \frac{\delta I}{\delta x}^2 \frac{\delta^2 I}{\delta y^2}}{\left(\frac{\delta I}{\delta x}^2 + \frac{\delta I}{\delta y}^2\right)^{\frac{3}{2}}}.$$

where, for example, $\delta I/\delta x$ is the first-order derivative of the intensity function I on the x-dimension. The distance to the center of the iris is found as the reciprocal of the aforementioned term. The orientation is calculated using the gradient, but its direction always indicates the highest change in luminance [see Fig. 1(a)]. The gradient is then multiplied by the inverse of the isophote curvature to disambiguate the direction of the center. Hence, the displacement vectors from every pixel to the estimated position of the centers, i.e., $D(x, y)$, are found to be

$$D(x,y) = -\frac{\left\{\frac{\delta I}{\delta x}, \frac{\delta I}{\delta y}\right\}\left(\frac{\delta I}{\delta x}^2 + \frac{\delta I}{\delta y}^2\right)}{\frac{\delta I}{\delta y}^2 \frac{\delta^2 I}{\delta x^2} - 2\frac{\delta I}{\delta x}\frac{\delta^2 I}{\delta x \delta y}\frac{\delta I}{\delta y} + \frac{\delta I}{\delta x}^2 \frac{\delta^2 I}{\delta y^2}}.$$

In this way, every pixel in the image gives a rough estimate of its own center, as shown in Fig. 1(b). Since the sign of the isophote curvature depends on the intensity of the outer side of the curve, bright and dark centers can be discriminated by the sign of the curvature. Since the sclera is assumed to be brighter than the cornea and the iris, votes with a positive isophote curvature are ignored as they are likely to come from noneye regions or highlights. In order to collect this information and deduce the location of a global eye center, $D(x, y)$ values are mapped into an accumulator [see Fig. 1(c)].

Instead of attributing the same importance to every center estimate, a relevance mechanism is used to yield more accurate center estimation, in which only the parts of the isophote following the edges of the object are considered. This weighting is performed by using the curvedness [23], i.e.,

$$\text{curvedness} = \sqrt{\frac{\delta^2 I}{\delta x^2}^2 + 2\frac{\delta^2 I}{\delta x \delta y}^2 + \frac{\delta^2 I}{\delta y^2}^2}.$$

The accumulator is then convolved with a Gaussian kernel so that each cluster of votes will form a single estimate. The maximum peak found in the accumulator is assumed to represent the location of the estimated eye center. An example is illustrated in Fig. 2. For this case, the eye center estimate can be clearly seen on the 3-D plot.

In [44], it is shown that the described method yields low computational cost allowing real-time processing. Furthermore, due to the use of isophotes, the method is shown to be robust against linear illumination changes and to moderate changes in the head pose. However, the accuracy of the eye center location significantly drops in the presence of head poses that are far from frontal. This is due to the fact that, in these cases, the analyzed eye structure is not symmetric, and thus, the algorithm delivers increasingly poor performance with respect to the distance from the frontal pose. This observation shows that it is desirable to be able to correct the distortion given by the pose so that the eye structure under analysis keeps the symmetry properties. To obtain the normalized image patches invariant to changes in the head pose, a head pose estimation algorithm will be employed.

B. Head Pose Estimation

Throughout the years, different methods for head pose estimation have been developed. The 3-D-model-based approaches achieve robust performance and can deal with large rotations. However, most of the method reasonably work in restricted domains only, e.g., some systems only work when there is stereo data available [29], [37], when there is no (self-) occlusion, or when the head is rotating not more than a certain degree [9]. Systems that solve most of these problems do not usually work in real time due to the complex face models that they use [50] or require accurate initialization. However, if the face model complexity is reduced to a simpler ellipsoidal or cylindrical shape, this creates a prospect for a real-time system and can be simply initialized starting from eye locations. The cylindrical head model (CHM) approach has been used by a number of authors [7], [9], [49]. Among them, the implementation of Xiao *et al.* [49] works remarkably well. This cylindrical approach is still capable of also tracking the head in situations where the head turns more than 30° from the frontal position and will be therefore used in this paper and outlined as follows.

To achieve good tracking accuracy, a number of assumptions are considered for the simplification of the problem. First of all, camera calibration is assumed to be provided beforehand, and a single stationary camera configuration is considered. For perspective projection, a pin hole camera model is studied.

The initial parameters of the CHM and its initial transformation matrix are computed as follows: Assuming that the face of the subject is visible and frontal, its size is used to initialize the cylinder parameters and pose $\mathbf{p} = [\omega_x, \omega_y, \omega_z, t_x, t_y, t_z]$ according to anthropometric values [13], [15], where ω_x, ω_y, and ω_z are the rotation parameters, and t_x, t_y, and t_z are the translation parameters. The eye locations are detected in the face region and are used to give a better estimate of t_x and t_y. Depth t_z is adjusted by using the distance between the detected eyes d. Finally, since the detected face is assumed to be frontal, the

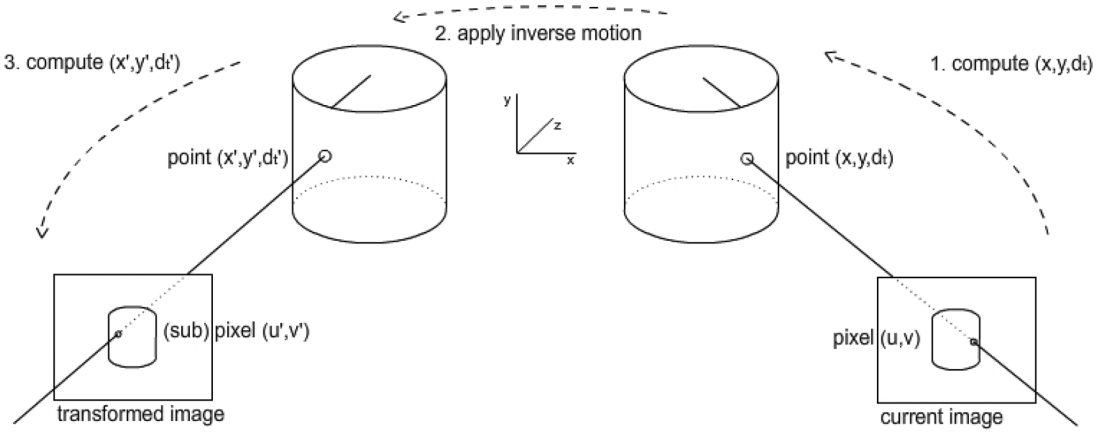


Fig. 3. Orientation of the cylinder and its visualization on the image plane.

initial pitch (ω_x) and yaw (ω_y) angles are assumed to be null, whereas the roll angle ω_z is initialized by the relative position of the eyes.

To analyze the effect of the motion of the CHM on the image frame, the relation between the 3-D locations of the points on the cylinder and their corresponding projections on the 2-D image plane need to be established. Therefore, the 3-D locations of the points with respect to the reference frame need to be determined first. This is obtained by sampling points on the cylinder. After obtaining the coordinates of these points on the 3-D elliptic cylindrical model, perspective projection is applied to get the corresponding coordinates on the 2-D image plane.

Since the CHM is assumed to be aligned along the $\mathbf{y}$-axis of the reference frame and to be positioned such that the center coincides with the origin (as shown in Fig. 3), any point $p = (p_x, p_y, p_z)^T$ on the cylinder satisfies the following explicit equation:

$$\left(\frac{p_x}{\mathbf{r_x}}\right)^2 + \left(\frac{p_z}{\mathbf{r_z}}\right)^2 = 1 \tag{1}$$

where $\mathbf{r_x}$ and $\mathbf{r_z}$ stand for the radii of the ellipse along the $\mathbf{x}$- and $\mathbf{z}$-axes, respectively. To calculate the coordinates of the points on the visible part of the cylinder, the front region is sampled in an $N_s \times N_s$ gridlike structure on the $x-y$ plane, and corresponding depth values are obtained by using (1). These sampled points are considered to summarize the motion of the cylinder, and they are employed in the Lukas–Kanade optical-flow algorithm. The perspective projection of the 3-D points on the elliptic cylindrical face model gives the 2-D pixel coordinates in the image plane. Let point $p = (p_x, p_y, p_z)^T$ in Fig. 3 be a point sampled on the cylinder and point $u = (u_x, u_y)^T$ be its projection on the image plane. Fig. 4 illustrates the side view of this setting by making a pin hole camera assumption for the sake simplification. Using the similarity of triangles in Fig. 4, the following equations apply for the relation between p and u:

$$p_x = \frac{p_z u_x}{\text{fl}}$$
$$p_y = \frac{p_z u_y}{\text{fl}} \tag{2}$$

where fl stands for the focal length of the camera. This relation is summarized by a perspective projection function $\mathbf{P}$, which

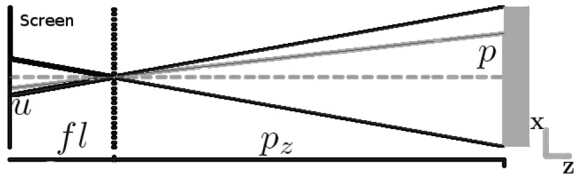


Fig. 4. Perspective projection of point p onto image plane by a pin hole camera assumption.

maps the 3-D points onto the 2-D image plane employing the previously given identities, i.e.,

$$\mathbf{P}(p) = u.$$

As shown in Fig. 3, the cylinder is observed at different locations and with different orientations at two consecutive frames F_i and F_{i+1}. This is expressed as an update in the pose vector $\mathbf{p_i}$ by the rigid motion vector $\Delta\mu_i = [\omega_x^i, \omega_y^i, \omega_z^i, \tau_x^i, \tau_y^i, \tau_z^i]$. To compute this motion vector, it is required to establish the relation between p_i and u_i of F_i and their corresponding locations on F_{i+1}. In the formulation of this relation, three transformation functions are employed, as illustrated in Fig. 3. The 3-D transformation function $\mathbf{M}$ maps p_i to p_{i+1}, whereas the 2-D transformation function $\mathbf{F}$ maps u_i to u_{i+1}, and the perspective projection function $\mathbf{P}$ maps p_i to u_i.

It can be derived that the explicit representation of the perspective projection function in terms of the rigid motion vector parameters and the previous coordinates of the point is [49]

$$\mathbf{P}\left(\mathbf{M}(p_i, \Delta\mu)\right) = \begin{bmatrix} p_i^x - p_i^y\omega_z + p_i^z\omega_y + \tau_x \\ p_i^x\omega_z + p_i^y - p_i^z\omega_x + \tau_y \end{bmatrix} \times \frac{\text{fl}}{-p_i^x\omega_y + p_i^y\omega_x + p_i^z + \tau_z}.$$

In the next section, the estimated head pose will be used to obtain the pose normalized eye patches.

III. Synergetic Eye Location and CHM Tracking

As mentioned in the previous section, the CHM pose tracker and the isophote-based eye location estimation methods have advantages over other reported methods. However, taken

separately, they cannot adequately work under certain circumstances. In [44], the eye region is assumed to be frontal so that the eye locator can use curved isophotes to detect circular patterns. However, since the method is robust to slight changes in the head pose, the system can be still applied with head poses up to $> 30^\circ$ at the cost of accuracy. On the other hand, the CHM pose tracker may erroneously converge to local minima and, after that, may not able to recover the correct track. By integrating the eye locator with the CHM, we aim to obviate these drawbacks.

Instead of a sequential integration of the two systems, an early integration is proposed. Relevant to this paper is the approach proposed in [40]. The authors combine a CHM with an AAM approach to overcome the sensitivity to large pose variations, initial pose parameters, and problems of re-initialization. In the same way, we make use of the competent attributes of the CHM, together with the eye locator proposed in [44], to broaden the capabilities of both systems and to improve the accuracy of each individual component. By comparing the transformation matrices independently suggested by both systems, in our method, the eye locations will be detected given the head pose, and the head pose will be adjusted given the eye locations. To this end, after the cylinder is initialized in 3-D space, the 2-D eye locations detected in the first frame are used as reference points (e.g., the "+" markers in Fig. 7). These reference points are projected onto the CHM so that the depth values of the eye locations are known. The reference eye points are then used to estimate the successive eye locations and are, in turn, updated by using the average of the found eye locations.

A. Eye Location by Pose Cues

Around each reference point projected onto the 3-D model, an area is sampled and transformed by using the transformation matrix obtained by the head pose tracker (see Fig. 5). The pixels under these sampled points are then remapped into a normalized canonical view (see Fig. 6). Note that extreme head poses are also successfully corrected, although some perspective projection errors are retained. The eye locator described in Section II-A is then applied to these pose-normalized eye regions. The highest peak in the obtained accumulator, which is closer to the center of the sampled region (therefore closer to the reference eye location obtained by pose cues), is selected as the estimated eye center (the white dots in Fig. 6 and the "x" markers in Fig. 7). In this way, as long as the CHM tracker is correctly estimating the head pose, the localized eyes can be considered to be optimal. Fig. 7 shows two examples in which the default eye locator would fail ("·" marker), but the pose-normalized eye estimation would be correct ("x" marker).

B. Pose Estimation by Eye Location Cues

Since there is uncertainty about the quality of the pose obtained by the head tracker, the found pose-normalized eye location can be used as a cue for quality control. Given that the 3-D position of the eyes is known, it is possible to calculate its pose vector and compare it with the one obtained by the head tracker. When the distance between the two pose vectors is larger than a certain threshold, the vectors are averaged, and the transformation matrix of the tracker is recomputed. In this way, the head model is adjusted to a location that should ease the correct convergence and therefore recover the correct track. As an additional quality control, the standard eye locator is constantly used to verify that the eye location found by pose cues is consistent with the one obtained without pose cues. Therefore, as in [30], when reliable evidence (e.g., the eye location in a frontal face) is collected and found to be in contrast with the tracking procedure, the latter is adjusted to reflect this.

In this manner, the eye locations are used to both initialize the cylinder pose and update it in case it becomes unstable, whereas the pose-normalized eye locations are used to constantly validate the tracking process. Therefore, the CHM tracker and the eye locator interact and adjust their own estimations by using each other's information. This synergy between the two systems allows for an initialization-free and self-adjusting system. A schematic overview of the full system is shown in Fig. 8, while its pseudocode is presented in Algorithm 1.

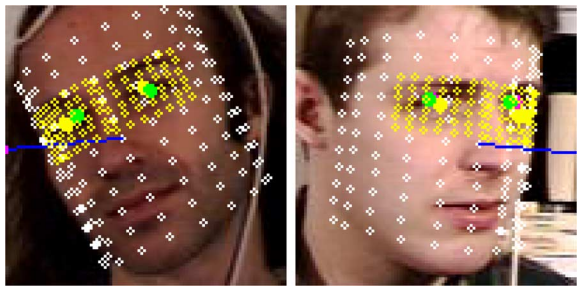


Fig. 5. Examples of eye regions sampled by the pose (yellow dot meshes).

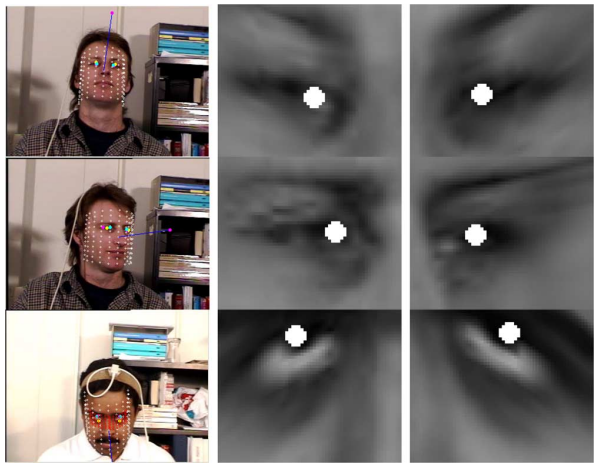


Fig. 6. Examples of extreme head poses and the respective pose-normalized eye locations. The results of the eye locator in the pose normalized eye region is represented by a white dot.

Fig. 7. Erroneous eye locations detected by $(\cdot)$ the standard eye locator, corrected by (x) the pose cues, according to $(+)$ the reference points.

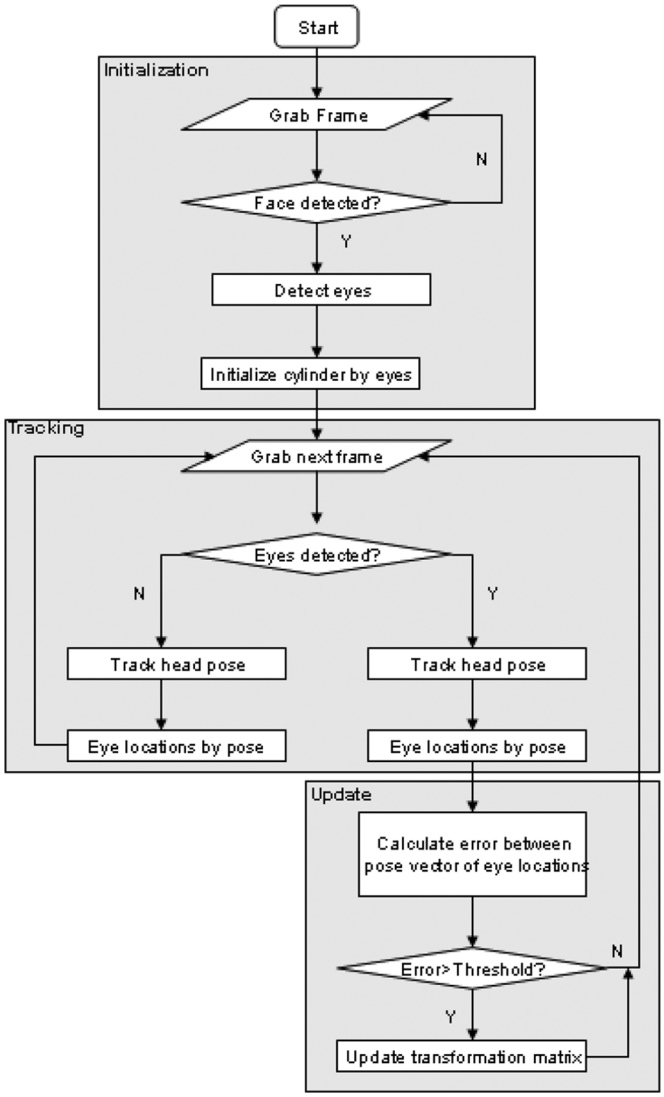


Fig. 8. Schematic diagram of the components of the system.

Algorithm 1 Pseudocode of estimating eye locations by head pose

Initialize parameters

— Detect face, and initialize cylinder parameters
— Get reference eye regions R_r and R_l.
— Use the distance between the eyes to get the depth element t_z.
— Initialize pose $\mathbf{p}$ using eye locations.

Iterate through all the frames

for $t = 0$ to last frame number **do**
— Assume that intensity is constant between consecutive frames, i.e., $I_{t+1} = I_t$.
— Compute gradient $\bigtriangledown I_{t+1}$ and the corresponding Gaussian pyramid for the current frame.
— Initialize pose to the previous pose $p_{t+1} = p_t$.

For all levels of Gaussian Pyramid

for $l = 0$ to 2 **do**
— Calculate motion between two frames $\mathbf{m} = \mathbf{p_{t+1}} * \mathbf{p_t^{-1}}$.
— Load Gaussian pyramid image $I(l)$.
— Initialize $\Delta\vec{p} = [[0,0,0,0,0,0]]$.

while maximum iterations not reached or $\Delta\vec{p} <$ threshold **do**
— Transform pixels p of $I(l)$ to p' with transformation matrix $\mathbf{M}$ and parameters $\mathbf{p}$ to compute $I_t(\mathbf{p})$.
— Update and scale face region boundaries $(\vec{u}, \vec{v})$.
— Do ray tracing to calculate t_z for each $p \in (\vec{u}, \vec{v})$.
— Apply perspective projection $p_x = \vec{u}_n * t_z$ and $p_y = \vec{v}_n * t_z$.
— Use inverse motion $\mathbf{m}'$ to get from p to p'.
— With back projection, calculate pixels (u', v').
— Compute I_t with $I_{t+1}(\mathbf{m}) - I_t(\mathbf{m}')$.
— Compute $\bigtriangledown I_{t+1}(\mathbf{m})(\partial\mathbf{T}/\partial\mathbf{p})$, where T summarizes the projection model.
— Compute the Hessian matrix in
— Compute $\sum w[\bigtriangledown I_{t+1}(\partial T/\partial p)]^T \sum[I_t - I_{t+1}]$.
— Compute $\Delta\vec{p}$ using
— Update the pose and the motion, i.e.,
— $\mathbf{p_{t+1}} = \Delta\vec{p} \circ \mathbf{p_{t+1}}$
— $\mathbf{m} = \Delta\vec{p} \circ \mathbf{m}$

end while
— Update transformation matrix $\mathbf{M} = \Delta\vec{p} \circ \mathbf{M}$.
— Transform reference eye regions R_r and R_l using $\mathbf{M}$.
— Remap eye regions to the pose-normalized view.
— Compute displacements vectors D on pose-normalized eye regions accordingly to [44] using

$$D(x,y) = -\frac{\left\{\frac{\delta I}{\delta x}, \frac{\delta I}{\delta y}\right\}\left(\frac{\delta I}{\delta x}^2 + \frac{\delta I}{\delta y}^2\right)}{\frac{\delta I}{\delta y}^2\frac{\delta^2 I}{\delta x^2} - 2\frac{\delta I}{\delta x}\frac{\delta^2 I}{\delta x \delta y}\frac{\delta I}{\delta y} + \frac{\delta I}{\delta x}^2\frac{\delta^2 I}{\delta y^2}}.$$

— Vote for centers weighted by $\sqrt{(\delta^2 I/\delta x^2)^2 + 2(\delta^2 I/\delta x \delta y)^2 + (\delta^2 I/\delta y^2)^2}$.
— Select isocenter closer to the center of the eye region as the eye estimate.
— Remap the eye estimate to cylinder coordinates.
— Create the pose vector from the eye location and compare it with the head tracker's.

if distance between pose vector $>$ threshold **then** average pose vectors and create new $\mathbf{M}$

end if

end for

end for

IV. Visual Gaze Estimation

In the previous section, we described how the 2-D eye center locations detected in the first frame are used as reference points (see Fig. 12). These reference points are projected onto the

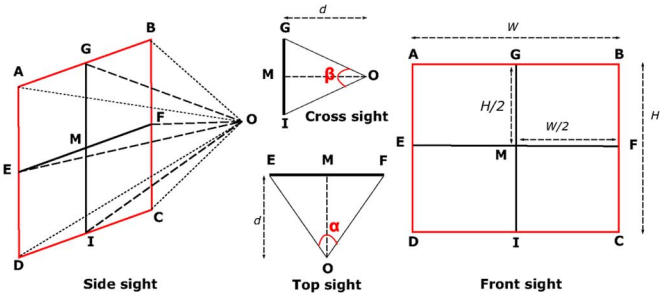


Fig. 9. Representation of the visual field of view at distance d.

CHM and are then used to estimate the successive pose-normalized eye center locations. Here, the displacement vectors between the resting position of the eyes (reference points) and the estimated eye location will be used to obtain joint visual gaze estimation, constrained within the visual field of view defined by the head pose.

A. Human Visual Field of View

Studies on the human visual field of view [34] show that, while looking straight ahead, it has a vertical span of 130° (60° above and 70° below) and approximately 90° on each side, which corresponds to a photographic objective angle of 180°.

The common field of view of the two eyes is called the binocular field of view and spans 120°. It is surrounded by two monocular fields of view of approximately 30°.

The field of view can be approximated by pyramid $OABCD$, where O represents the point between the two eyes and rectangle $ABCD$ represents the visual field of view at distance d. Furthermore, angles α and β denote the horizontal and vertical angles of the visual field of view in binocular human vision, respectively [25]. Since the pyramid is an approximation of the field of view, we are able to center it on the gaze point M so that it is in the middle of the field of view. In this case, vector OM denotes the visual gaze vector (see Fig. 9).

Width W and height H of the visual field at distance d are computed by

$$W = 2MF = 2d\tan\frac{\alpha}{2} \qquad H = 2MG = 2d\tan\frac{\beta}{2}.$$

The projection of the visual field of view on the gazed scene in front of a user is quadrilateral $A'B'C'D'$ with the central gaze point M', and it is calculated by the intersection between the plane of the target scene $P : ax + by + cz + d = 0$ and lines (OA), (OB), (OC), (OD), and (OM). The head pose parameters computed by the method described in Section II-B are used to define the projection of the region of interest in the target scene.

B. Pose-Retargeted Gaze Estimation

So far, we considered the visual field of view defined by the head pose only, modeled so that the visual gaze of a person (the vector defining the point of interest) corresponds to the middle of the visual field of view. However, it is clear that the displacements of the eyes from their resting positions will influence the estimation of the visual field of view.

In general, most methods avoid this problem by assuming that the head does not move at all and assume that the eyes do not rotate in the ocular cavities but just shift on the horizontal and vertical planes [45]. In this way, the problem of eye displacement is simply solved by a 2-D mapping of the location of the pupil (with respect to an arbitrary anchor point) and known locations on the screen. The mapping is then used to interpolate between the known target locations in order to estimate the point of interest in the gazed scene. This approach is often used in commercial eye trackers, using high-resolution images of the eyes and IR anchor points. However, this approach forces the user to use a chin rest to avoid head movements, which will results in wrong mappings.

In this paper, instead of focusing on modeling the shape of the eyes or the mapping between their displacement vectors, we make the assumption that the visual field of view is only defined by the head pose and that the point of interest (defined by the eyes) does not fall outside the head-pose-defined field of view. This assumption corresponds to the study in [39], where it is shown that the head pose contributes to about 70% of the visual gaze. Here, we make the observation that the calibration step is not directly affected by the head position. For example, when the calibration is performed while the head is slightly rotated and/or translated in space, the mapping is still able to compute the gazed location by interpolating between known locations (as long as the head position does not vary). In this way, the problem of 3-D gaze estimation is reduced to the subproblem of estimating it in 2-D (e.g., using eyes only), removing the constraints on head movements.

Instead of learning all possible calibrations in 3-D space, we propose to automatically retarget a set of known points on a target plane (e.g., a computer screen) in order to simulate recalibration each time the user moves his/her head. In fact, if the known points are accordingly translated to the parameters obtained from the head pose, it is possible to use the previously obtained displacement vectors and recalibrate using the new known points on the target plane. To this end, a *calibration plane* is constructed, which is attached to the front of the head as in Fig. 10(a), so that it can be moved using the same transformation matrix obtained from the head pose estimator (to ensure that it accordingly moves). The calibration plane is then populated during the calibration step, where the user is requested to look at a known set of points on the target plane. The ray between the center of the head and the known point on the target plane is then traced until the calibration plane is intersected. In this way, the relation between the calibration plane and the target plane (e.g., a computer screen) is also computed.

Since the calibration points are linked to the head-pose-constructed visual field of view, their locations will change when the head moves in the 3-D space in front of the target plane. Hence, every time that the head moves, the intersection points between the ray going from the anchor point to the calibration point are computed in order to construct the new set of known points on the target plane. Using this new set of known points and the known pose-normalized displacement vectors as collected during the calibration phase, it is possible to automatically recalibrate and learn a new mapping. Fig. 10(b) shows how the calibration points are projected on the calibration plane,

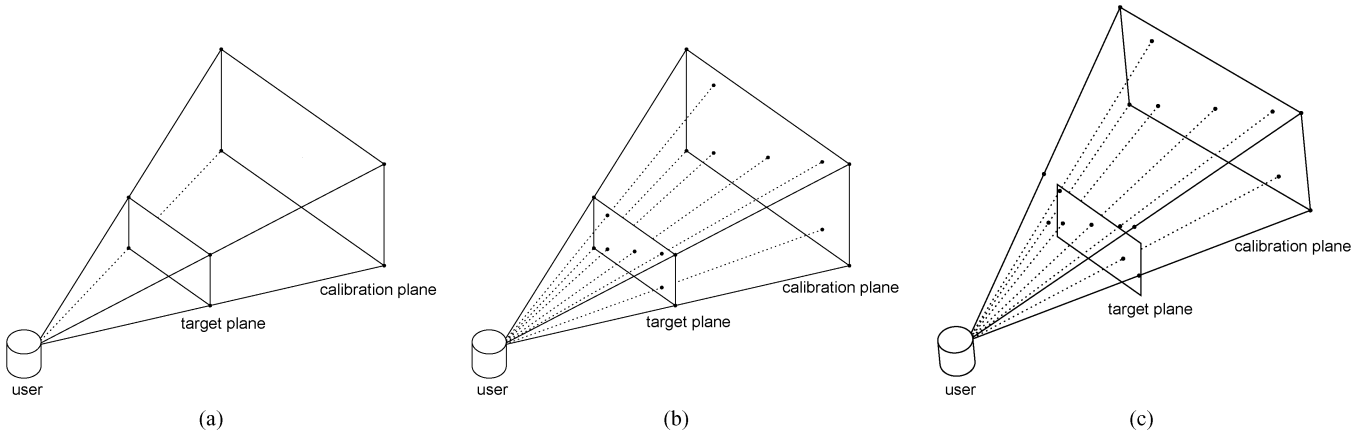


Fig. 10. (a) The construction of the calibration plane, (b) the intersection of the calibration points on known target plane points, and (c) the effect on the known points on the target plane while moving the head in the 3-D space.

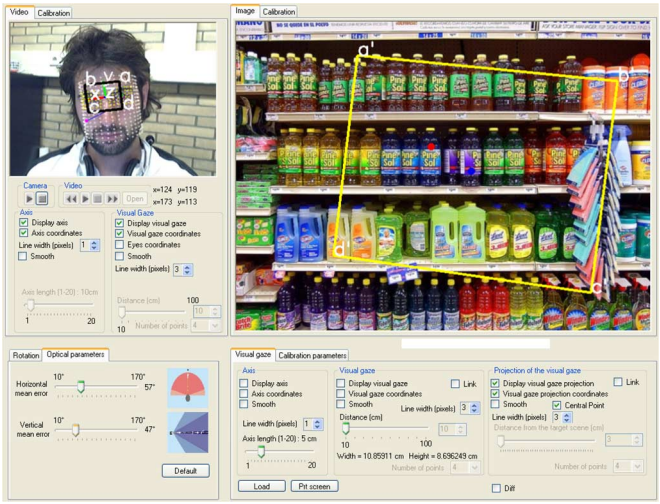


Fig. 11. Screenshot of the final system.

and Fig. 10(c) illustrates how these points change during head movements, obtaining new intersections on the target plane (the *pose-retargeted* known points).

Fig. 11 shows a screenshot of the final system, working in real time using a simple webcam. The quadrilateral indicates the user's region of interest (defined by head pose only), while the red dot represent the point of interest within it, obtained by the proposed system.

V. Experiments

In this paper, the three components need an independent evaluation, i.e., the accuracy provided by the eye center location given the head pose, the accuracy obtained by the head pose estimation given the eye center location, and the accuracy of the combined final visual gaze estimation. In the following sections, the data sets, the error measures, and the result for each of three components are described and discussed.

A. Eye Location Estimation

The performance obtained by using head pose cues in eye location are evaluated using the Boston University head pose database [9]. The database consists of 45 video sequences, where five subjects were asked to perform nine different head motions under uniform illumination in a standard office setting. The head is always visible, and there is no occlusion except for some minor self-occlusions. Note that the videos are in low resolution (320×240 pixels); hence, the iris diameter roughly corresponds to 4 pixels.

A Flock of Birds tracker records the pose information coming from the magnetic sensor on the person's head. This system claims a nominal accuracy of 1.8 mm in translation and 0.5° in rotation. However, La Cascia *et al.* [9] have experienced a lower accuracy due to the interfering electromagnetic noise in the operating environment. Nonetheless, the stored measurements are still reliable enough to be used as the ground truth. As no annotation of the eye location on this data set is available, we manually annotated the eyes of the subjects on 9000 frames. These annotations are publicly available in [43].

In quantifying the error, we used the 2-D *normalized error*. This measure was introduced in [20] and is widely used in the eye location literature [6], [8], [16], [42], [51]. The normalized error represents the error obtained by the worse eye estimation and is defined as

$$e = \frac{\max(d_{\text{left}}, d_{\text{right}})}{d} \tag{3}$$

where d_{left} and d_{right} are the Euclidean distance between the located eyes and the ones in the ground truth, and d is the Euclidean distance between the eyes in the ground truth. For this measure, $e \leq 0.25$ (a quarter of the interocular distance) roughly corresponds to the distance between the eye center and the eye corners, $e \leq 0.1$ corresponds to the range of the iris, and $e \leq 0.05$ corresponds the range of the cornea. In order to give upper and lower bounds to the accuracy, in Fig. 12 we also show the *minimum normalized error*, obtained by considering the best eye estimation only.

The accuracy achieved by the proposed unified approach is presented in Fig. 12, together with the baseline accuracy obtained by the standard eye locator [44]. In the latter, the approximate face position is estimated using the boosted cascade face detector proposed in [47], where the rough positions of the left and right eye regions are estimated by anthropometric

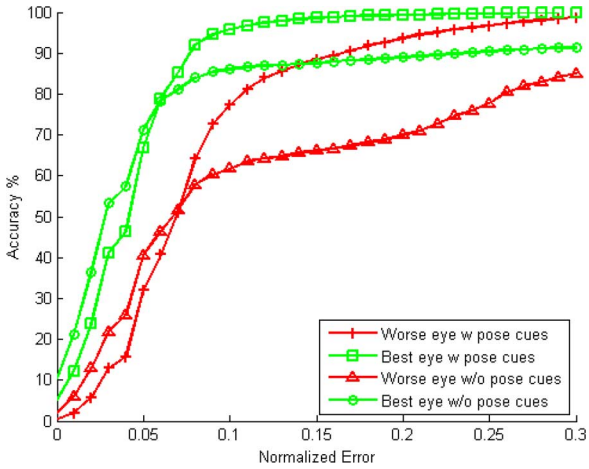


Fig. 12. Comparison between the eye detection results with and without pose cues.

TABLE I
EFFECT OF POSE CUES IN EYE LOCALIZATION

	Worse eye		Best eye	
	Without pose	With pose	Without pose	With pose
$e \leq 0.05$	40.6	31.93	66.78	71.27
$e \leq 0.1$	61.73	77.27	86.03	95.81
$e \leq 0.15$	66.14	88.46	87.87	98.6
$e \leq 0.2$	70	93.67	93.67	99.29
$e \leq 0.25$	77.72	96.74	96.74	99.73

relations [15]. For the cases in which the face cannot be detected, the maximum possible localization error is assigned, considering the limits of the detected face and anthropometric measures as follows. The maximum achievable error is assumed to be half of the interocular distance, which corresponds to 0.5. Therefore, a default error value of 0.5 is assigned to both eyes for the frames in which a face is not detected. In our experiments, the faces of the subjects were not detected in 641 frames, which corresponds to 7.12% of the full data set. The working range of the face detector is around 30° around each axis, while certain head poses in the data set are larger than 45°. The accuracy is represented in percentages for a normalized error of range [0, 0.3]. A performance comparison is provided for the best and worse eye location estimations, where certain precise values are also given in Table I for several normalized error values.

From Fig. 12, it is shown that the pose cues improve the overall accuracy of the eye detector. In fact, for an allowed error larger than 0.1, the unified scheme provides an improvement in accuracy from 16% to 23%. For smaller error values, the system performs slightly worse than the standard eye locator. The eye detection results obtained by using pose cues depict a significant overall improvement over the baseline results. However, we note a small drop in accuracy for precise eye location $(e \leq 0.05)$. This is due to interpolation errors occurring while sampling and remapping the image pixels to pose-normalized eye regions. In fact, as shown in Fig. 6, in specific extreme head poses, the sampled eye may not appear as completely circular shapes due to perspective projections. Therefore, the detection is shifted by one or two pixels. Given the low resolution of the videos, this shift can easily bring the detection accuracy beyond the $e \leq 0.05$ range. However, given the low resolution, this error is barely noticeable.

B. Head Pose Estimation

Since the ground truth is provided by the Boston University head pose database [9], it is also used to evaluate the effect of using eye location cues in head pose estimation. To measure the pose estimation error, the root-mean-square error (RMSE) and standard deviation (STD) values are used for the three planar rotations, i.e., ω_x, ω_y, and ω_z.

To measure the accuracy of the pose, two scenarios are considered. In the first scenario, the template is created from the first frame of the video sequence and is kept constant for the rest of the video; in the second scenario, the template is updated at each frame so that the tracking is always performed between two successive frames. Table II shows the improvement in the RMSE and the STD given by using eye location cues in both scenarios. Note that, without using the eye cues, the updated template gives the best results. On the other hand, if the eye cues are considered, the accuracy of the fixed template becomes better than the updated one. This due to the fact that using the eye cues while updating the template might introduce some errors at each update, which cannot be recovered at later stages. However, for both scenarios, the use of eye cues presents an improvement in the estimation of the pose angles. Some challenging examples of the results obtained by our implementation of the CHM head pose tracker are represented in Fig. 13 for challenging roll, yaw, and pitch rotations. The graphs with values for the ground truth and for the accuracy of the tracker for the respective videos are shown in Fig. 14. It can be derived that the system is able to cope with these extreme head poses.

In the last two columns of Table II, we compare our results with two other methods in the literature, which use the same database. Similar to our method, Sung *et al.* [40] propose a hybrid approach combining AAMs and cylinder head models to extend the operating range of the AAM. An and Chung [2] propose to replace the traditional CHM with a simple 3-D ellipsoidal model. They provide comparison of accuracy with planar and cylindrical models. Here, we consider the accuracy reported by Sung *et al.* and by An and Chung on the CHM [2]. From Table II, it is shown that our method provides comparable or better results with respect to the compared methods.

Hence, our experiments show that using eye cues has an overall positive effect on the average RMSE. However, it is important to note that enhancing the head tracker using the eye cues to fix the transformation matrix does not have a direct effect on the accuracy. The main effect is obtained by the re-initialization of the cylinder in a position that allows for a correct convergence once the pose tracker converges to a local minimum. In fact, by closely analyzing the results, it can be derived that, by using the eye cues, the accuracy of the pose is decreased for particular subjects showing extreme head poses.

This issue is related to the approach used to fix the transformation matrix. In our approach, we assume that the eye located given the correct pose are the correct ones, but this will not be true in the presence of highlights, closed eye, or very extreme head poses (e.g., when the head is turned by 90° and only one eye is visible). In these specific cases, averaging by the

TABLE II
COMPARISON OF THE RMSE AND THE STD

	Fixed template				Updated template				Sung	An
	With eye cues		Without eye cues		With eye cues		Without eye cues		*et al.* [41]	*et al.* [2]
	RMSE	STD	RMSE	STD	RMSE	STD	RMSE	STD	RMSE	RMSE
Pitch (ω_x)	5.26	4.67	6.00	5.21	5.57	4.56	5.97	4.87	5.6	7.22
Yaw (ω_y)	6.10	5.79	8.07	7.37	6.45	5.72	6.40	5.49	5.4	5.33
Roll (ω_z)	3.00	2.82	3.85	3.43	3.93	3.57	4.15	3.72	3.1	3.22

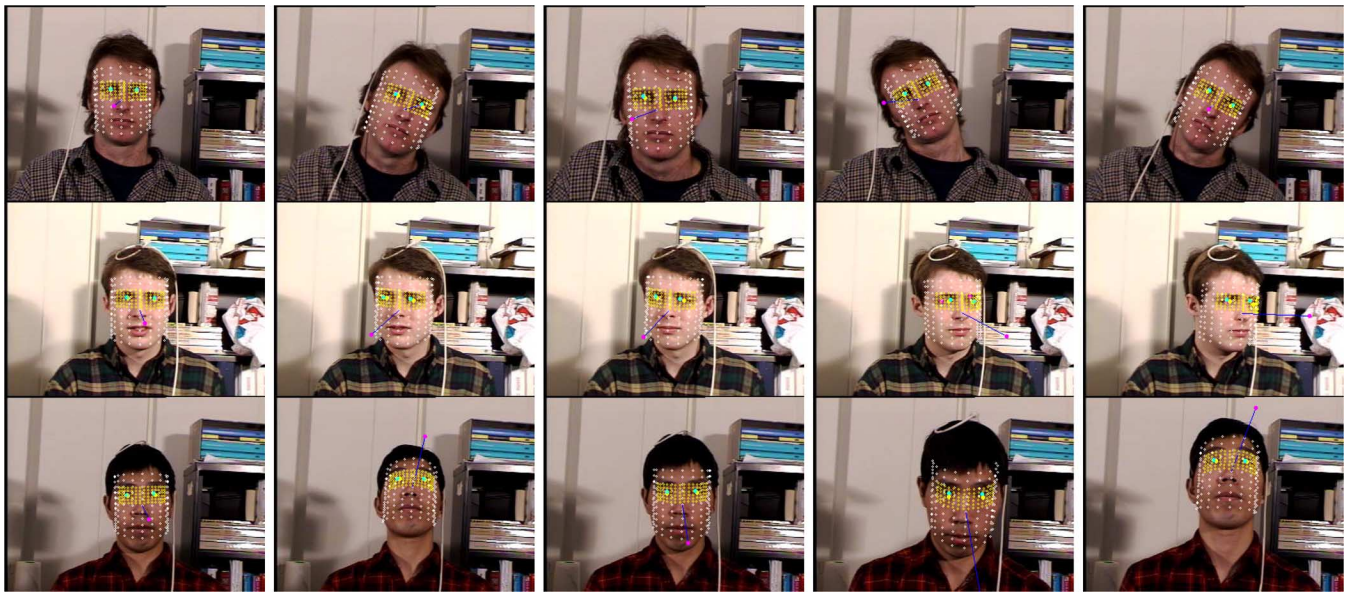


Fig. 13. Qualitative examples of result on roll, yaw, and pitch angles on videos showing extreme head poses.

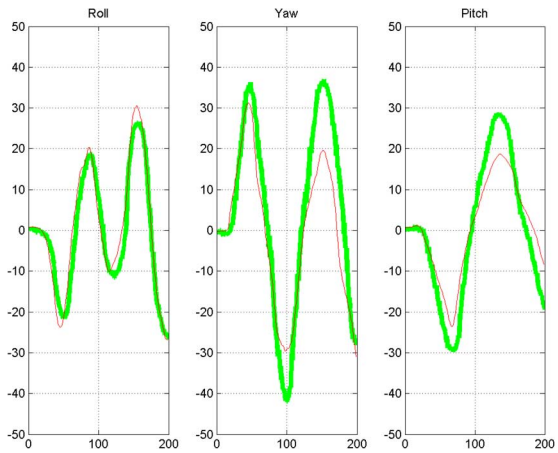


Fig. 14. Examples of quantitative result on roll, yaw, and pitch angles on videos showing extreme head poses. (Green) Ground truth. (Red) Tracking results.

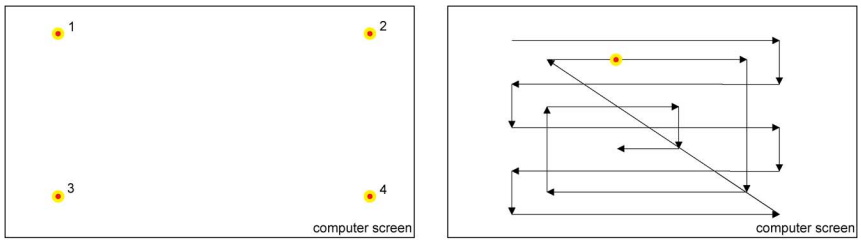
Fig. 15. Some of the subjects.

transformation matrix suggested by the eye location might negatively affect an otherwise correct transformation matrix given by the head tracker. Fortunately, the eye locator can be considered quite accurate, and therefore, these cases do not occur very often, and the track is recovered as soon as the difficult condition is resolved or a semifrontal face is detected again.

C. Visual Gaze Estimation

This section describes the experiments performed to evaluate the proposed gaze estimation system. To this end, a heterogeneous data set was collected, which includes 11 male and female subjects with different ethnicity, with and without glasses, and with different illumination conditions. Fig. 15 shows some examples of the subjects in the data set. The data were collected using a single webcam and without the use of a chin rest. The subject sits at a distance of 750 mm from the computer screen and the camera. The subject's head is approximately in the center of the camera image. The resolution of the captured images is 720 $\times$ 576 pixels, and the resolution of the computer screen is 1280 $\times$ 1024 pixels. To test the system under natural and extreme head movements, the subjects were requested to perform two set of experiments.

The first task, named *static dot gazing*, is targeted at evaluating how much the head pose can be compensated by the eye location. The subjects are requested to gaze with their eyes at a static point on the computer screen [see Fig. 16(a)] and move



(a) (b)

Fig. 16. Two tasks performed during the data collection. (a) Static dot gazing. (b) Dot following.

TABLE III
MEAN PIXEL ERROR AND STD COMPARISON ON THE STATIC DOT GAZING TASK

Subject	Eyes Only		Pose-Normalized		Pose-Retargeted	
	Mean	Std	Mean	Std	Mean	Std
1	688.91, 281.29	1090.71, 552.12	338.97, 366.42	181.04, 280.57	186.11, 223.51	191.32, 212.11
2	633.08, 187.74	464.23, 143.99	310.17, 226.07	182.89, 201.83	134.91, 191.77	197.01, 153.41
3	2285.11, 301.45	4929.71, 436.26	359.97, 246.45	184.91, 216.05	161.21, 255.71	201.04, 168.91
4	1202.11, 2664.01	2537.21, 7280.43	346.56, 330.79	164.11, 246.59	190.12, 164.84	199.37, 154.58
5	1388.72, 276.79	1073.91, 630.76	428.04, 327.46	222.63, 287.75	239.25, 215.67	200.78, 225.44
6	874.85, 239.81	726.24, 491.89	429.17, 265.61	215.31, 190.13	232.84, 234.02	175.45, 177.93
7	710.24, 328.63	443.08, 224.25	449.44, 217.21	240.39, 175.33	242.87, 162.01	188.21, 140.37
8	666.58, 257.17	376.31, 194.65	397.56, 236.25	226.65, 181.82	196.38, 152.45	191.22, 131.81
9	623.68, 316.73	412.11, 209.09	395.59, 337.47	206.61, 246.04	204.87, 220.94	197.46, 202.01
10	750.93, 332.06	947.91, 1462.83	430.95, 319.41	247.44, 263.63	272.23, 223.17	231.42, 205.61
11	924.62, 398.24	2297.01, 297.93	443.03, 580.46	229.99, 412.91	252.93, 320.81	186.16, 255.22

their head around while still looking at the specific point. The point is displayed at certain locations on the screen for about 4 s each time. When the point is displayed on the screen, the subject is asked to look at it and then to rotate his/her head toward the point's location. When the desired head position is reached the subjects are asked to move their head while their eyes are still gazing at the displayed point. The location and the order in which the points are displayed are shown in Fig. 16(a). The second task, named *dot* following, is targeted at evaluating the gaze estimation performance while following a dot on the screen. The test subjects are requested to look and follow a moving point on the screen in a natural way, using their eyes and head if required. The path followed by the dot is shown in Fig. 16(b).

The ground truth is collected by recording the face of the subject and the corresponding on-screen coordinates where the subjects are looking.

In order to test the performance of the proposed approach, three different methods are tested.

1) **Eyes-only gaze estimator**: This estimator uses the anchor–pupil vectors directly into the mapping as in the system proposed in [45]. Hence, when the user moves his head from the calibration position, the mapping is bound to fail. This experiment is performed to evaluate the limitations of the classic mapping approaches in the presence of head movements.
2) **Pose-normalized gaze estimator**: This estimator uses the information about the position of the user's head to pose normalize the anchor–pupil vectors. During the calibration step, the displacement vectors between the anchor point and the location of the eyes are calculated from the pose-normalized CHM. These vectors are used together with the coordinates of the corresponding points on the computer screen for training. Then, the estimator approximates the coefficients of the underlying model by minimizing the error measure of the misfit of the generated estimates by a candidate model and the train data. When a certain threshold is reached, the model is accepted and used for the estimation of the point of interest when a future displacement vector is constructed;
3) **Pose-retargeted gaze estimator**: This is the approach proposed in Section IV-B, which treats the 3-D gaze estimation problem as a superset of 2-D problems. Moreover, this estimator uses pose-normalized displacement vectors. The main difference between the pose-retargeted estimator and the pose-normalized one is that, when the user moves his/her head, the set of known points is retargeted using the head pose information. The new coefficients of the underlying model are then approximated and used for the estimate of the new point of interest.

Table III shows the mean errors of the three gaze estimators in the first task (static dot gazing) for each of the tested subjects. Due to the big changes in head pose while keeping the eyes fixed, the eyes-only estimator has a significantly larger error and STD with respect to the other methods, which include pose-normalized displacement vectors. The pose-normalized estimator, in fact, has a mean error of (393.58, 313.96) pixels, corresponding to an angle of (8.5°, 6.8°), whereas the pose-retargeted estimator has a mean error of (210.33, 214.99) pixels, corresponding to an angle of (4.6°, 4.7°) in the x- and y-direction, respectively. The proposed pose-retargeted estimator improves the method with a factor of approximately 1.87 in x-direction and a factor of about 1.46 in y-direction, as compared with the pose-normalized system.

Table IV shows the results of the second task (dot following). In this task, due to the fact that the head significantly shifts

TABLE IV
MEAN PIXEL ERROR AND STD COMPARISON ON THE DOT FOLLOWING TASK

Subject	Eyes Only		Pose-Normalized		Pose-Retargeted	
	Mean	Std	Mean	Std	Mean	Std
1	3461.68, 938.55	1931.83, 567.68	238.88, 112.91	159.53, 69.42	75.95, 117.01	71.02, 76.82
2	3125.42, 361.55	5874.41, 253.68	229.78, 104.72	137.44, 73.58	79.16, 115.87	58.79, 82.01
3	3531.19, 564.11	7725.46, 353.82	253.51, 103.87	162.11, 77.08	78.73, 128.01	67.48, 77.07
4	2380.21, 400.89	2002.35, 608.31	277.71, 134.53	180.27, 139.32	99.29, 115.16	108.59, 150.38
5	3554.94, 656.51	2799.42, 468.44	268.51, 105.09	165.54, 77.78	85.77, 101.13	78.03, 72.79
6	2365.84, 472.37	1574.86, 336.32	254.63, 95.47	165.61, 62.83	80.25, 78.27	78.67, 57.13
7	3606.85, 729.86	3414.06, 1730.97	282.74, 101.62	179.79, 76.36	93.81, 104.64	84.21, 81.32
8	3332.96, 573.66	6989.07, 625.28	278.79, 188.84	189.94, 137.01	92.25, 92.77	85.31, 65.44
9	11958.67, 775.88	21508.32, 752.94	250.25, 200.03	180.95, 139.38	92.52, 99.32	72.27, 71.75
10	5082.26, 731.92	9972.58, 719.33	295.22, 168.61	190.91, 115.41	91.92, 92.21	86.09, 59.38
11	8482.27, 693.31	13728.11, 832.69	303.83, 179.69	201.23, 162.51	89.36, 98.12	109.38, 138.84

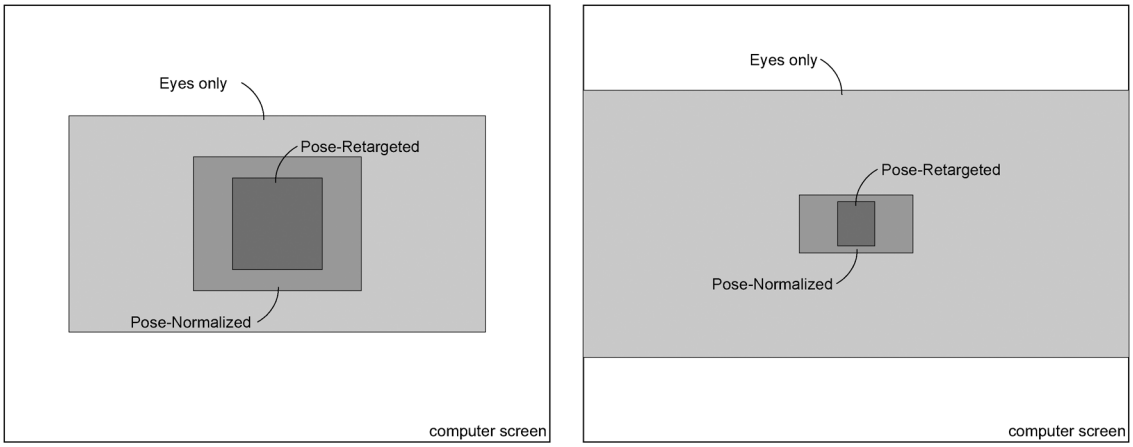


Fig. 17. Errors for the three tested estimators compared with the computer screen for (a) the static dot gazing task and (b) the dot following task.

from the calibration position to allow the eyes to comfortably follow the dot on the screen, the mapping in the eyes-only estimator completely fails. However, the pose-normalized estimator achieves a mean error of (266.71, 135.94) pixels, which corresponds to an angle of (5.8°, 3.0°), while the pose-retargeted estimator has a mean error of (87.18, 103.86) pixels, corresponding to an angle of (1.9°, 2.2°) in the x- and y-direction, respectively. Note that this error is significantly smaller than the previous task due to the fact that, here, the head naturally moves with respect to the eyes. When compared with the pose-normalized estimator, the pose-retargeted estimator improves the accuracy with a factor of approximately 3.05 in x-direction and with a factor of about 1.31 in y-direction. The differences between the accuracy obtained by the different systems in both tasks is visually represented in Fig. 17.

Although the average error obtained by the proposed system seems high at first, one should consider that the human fovea covers $\sim 2^\circ$ of the visual field, in which everything can be seen without requiring a saccade. Therefore, when asking a subject to gaze at a specific location, there is always an inherent error on the gaze ground truth. In fact, assuming that the test subjects are sitting at a distance of 750 mm from the computer screen, the projection of the foveal error $\epsilon_f = 2^\circ$ on the target plane corresponds to a window of about 92×92 pixels, which is in the same magnitude as that of the results obtained by the proposed system. By analyzing the causes for the errors (and the big STD), we note that, in most cases, the results in the y-direction are worse than the results in x-direction. There are two main reasons for this: 1) The camera is situated on top of the computer screen; thus, when the test subject is gazing at the bottom part of the screen, the eyelids obscure the eye location, and significant errors are introduced by the eye locator. 2) The eyes move less in y-direction than in the x-direction. Furthermore, errors in the eye center locator seriously affect the system, as an error of just a few pixels on the eye estimation result in significant displacements at a distance of 750 mm.

However, it is clear that the proposed pose-retargeted estimator outperforms the other tested approaches in all the experiments, whereas the pose-normalized estimator clearly outperforms the method based on the eyes only. This clearly indicates that it is beneficial to combine head pose and eye information in order to achieve better, more natural, and accurate gaze estimation systems.

VI. CONCLUSION

In this paper, we have proposed a deep integration of a CHM-based head pose tracker and an isophote-based eye locator in a complementary manner, so that both systems can benefit from each other's evidence. Experimental results have shown that the accuracy of both independent systems is improved by their combination. The eye location estimation of the unified scheme achieved an improvement in accuracy from 16% to 23%, while

the pose error has been improved from 12% to 24%. Aside from the improvements in accuracy, the operating range of the eye locator has been extended (by more than 15°) by the head tracker, and the ineffectiveness of the previously reported eye location methods against extreme head poses has been compensated. Furthermore, automatic quality control and re-initialization of the head tracker have been provided by the integration of the eye locator, which helps the system in recovering to the correct head pose. Consequently, the proposed unified approach allows for an autonomous and self-correcting system for head pose estimation and eye localization. Finally, the information obtained by the proposed system has been combined in order to project the visual gaze of a person on the target scene by retargeting a set of known points using the head pose information. The evaluation using the collected data set has proven that the joint eye and head information results in a better visual gaze estimation, achieving a mean error between 2° and 5° on different tasks without imposing any restraints on the position of the head.

ACKNOWLEDGMENT

The authors would like to thank A. Lablack, Z. Yücel, and K. Stefanov for their valuable contributions to this work.

REFERENCES

[1] J. S. Agustin, J. P. Hansen, and J. Mateo, "Gaze beats mouse: Hands-free selection by combining gaze and emg," in *Proc. CHI*, 2008, pp. 3039–3044.

[2] K. H. An and M. Chung, "3D head tracking and pose-robust 2D texture map-based face recognition using a simple ellipsoid model," in *Proc. Intell. Robots Syst.*, 2008, pp. 307–312.

[3] S. Asteriadis, N. Nikolaidis, A. Hajdu, and I. Pitas, "An eye detection algorithm using pixel to edge information," in *Proc. Int. Symp. Control, Commun. Signal Process.*, 2006.

[4] S. Ba and J. Odobez, "A probabilistic framework for joint head tracking and pose estimation," in *Proc. ICPR*, 2004, pp. 264–267.

[5] S. Ba and J. Odobez, "From camera head pose to 3D global roomhead pose using multiple camera views," in *Proc. Int. Workshop Classification Events Activities Relationships*, 2007.

[6] L. Bai, L. Shen, and Y. Wang, "A novel eye location algorithm based on radial symmetry transform," in *Proc. ICPR*, 2006, pp. 511–514.

[7] L. Brown, "3D head tracking using motion adaptive texture-mapping," in *Proc. CVPR*, 2001, pp. I-998–I-1003.

[8] P. Campadelli, R. Lanzarotti, and G. Lipori, "Precise eye localization through a general-to-specific model definition," in *Proc. BMVC*, 2006, pp. 187–198.

[9] M. La Cascia, S. Sclaroff, and V. Athitsos, "Fast, reliable head tracking under varying illumination: An approach based on registration of texture-mapped 3D models," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 4, pp. 322–336, Apr. 2000.

[10] COGAIN, Communication by gaze interaction: Gazing into the future [Online]. Available: http://www.cogain.org 2006

[11] T. Cootes, G. Edwards, and C. Taylor, "Active appearance models," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 23, no. 6, pp. 681–685, Jun. 2001.

[12] D. Cristinacce, T. Cootes, and I. Scott, "A multi-stage approach to facial feature detection," in *Proc. BMVC*, 2004, pp. 277–286.

[13] N. A. Dogson, "Variation and extrema of human interpupillary distance," *Proc. SPIE*, vol. 5291, pp. 36–46, 2004.

[14] A. Doshi and M. M. Trivedi, "On the roles of eye gaze and head pose in predicting driver's intent to change lanes," *IEEE Trans. Intell. Transp. Syst.*, vol. 10, no. 3, 2009.

[15] C. C. Gordon, B. Bradtmiller, T. Churchill, C. E. Clauser, J. T. McConville, I. O. Tebbetts, and R. A. Walker, Anthropometric survey of us army personnel: Methods and summary statistics U.S. Army Natick Res., Natick, MA, Tech. Rep. NATICK/TR-89/044, 1988.

[16] M. Hamouz, J. Kittlerand, J. K. Kamarainen, P. Paalanen, H. Kalviainen, and J. Matas, "Feature-based affine-invariant localization of faces," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 9, pp. 1490–1495, Sep. 2005.

[17] D. W. Hansen and J. P. Hansen, "Eye typing with common cameras," in *Proc. Symp. Eye Track. Res. Appl.*, 2006, p. 55.

[18] Y. Hu, L. Chen, Y. Zhou, and H. Zhang, "Estimating face pose by facial asymmetry and geometry," in *Proc. Autom. Face Gesture Recog.*, 2004, pp. 651–656.

[19] K. Huang and M. Trivedi, "Robust real-time detection, tracking and pose estimation of faces in video streams," in *Proc. ICPR*, 2004, pp. 965–968.

[20] O. Jesorsky, K. J. Kirchberg, and R. Frischholz, "Robust face detection using the Hausdorff distance," in *Proc. Audio Video Biometric Pers. Authentication*, 1992, pp. 90–95.

[21] Q. Ji and X. Yang, "Real-time eye, gaze and face pose tracking for monitoring driver vigilance," *Real Time Imaging*, vol. 8, no. 5, pp. 357–377, Oct. 2002.

[22] S. Kim, S.-T. Chung, S. Jung, D. Oh, J. Kim, and S. Cho, "Multi-scale Gabor feature based eye localization," in *Proc. World Acad. Sci., Eng. Technol.*, 2007, pp. 483–487.

[23] J. Koenderink and A. J. van Doorn, "Surface shape and curvature scales," *Image Vis. Comput.*, vol. 10, no. 8, pp. 557–565, Oct. 1992.

[24] B. Kroon, A. Hanjalic, and S. M. Maas, "Eye localization for face matching: Is it always useful and under what conditions?," in *Proc. CIVR*, 2008, pp. 379–388.

[25] A. Lablack, F. Maquet, and C. Djeraba, "Determination of the visual field of persons in a scene," in *Proc. VISAPP*, Jan. 2008, pp. 313–316.

[26] S. R. Langton, H. Honeyman, and E. Tessler, "The influence of head contour and nose angle on the perception of eye-gaze direction," *Percept. Psychophys.*, vol. 66, no. 5, pp. 752–771, Jul. 2004.

[27] X. Liu, N. Krahnstoever, T. Yu, and P. Tu, "What are customers looking at?," in *Proc. AVSS*, 2007, pp. 405–410.

[28] Y. Matsumoto and A. Zelinsky, "An algorithm for real-time stereo vision implementation of headpose and gaze direction measurement," in *Proc. FG*, 2000, pp. 499–505.

[29] L. Morency and T. Darrell, "Head gesture recognition in intelligent interfaces: The role of context in improving recognition," in *Proc. IUI*, 2006, pp. 32–38.

[30] L. Morency, A. Rahimi, and T. Darrell, "Adaptive view based appearance models," in *Proc. CVPR*, 2003, pp. I-803–I-810.

[31] E. Murphy-Chutorian and M. Trivedi, "Head pose estimation in computer vision: A survey," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 4, pp. 607–626, Apr. 2009.

[32] R. Newman, Y. Matsumoto, S. Rougeaux, and A. Zelinsky, "Real-time stereo tracking for head pose and gaze estimation," in *Proc. Autom. Face Gesture Recog.*, 2000, pp. 122–128.

[33] Z. Niu, S. Shan, S. Yan, X. Chen, and W. Gao, "2D cascaded adaboost for eye localization," in *Proc. ICPR*, 2006, pp. 1216–1219.

[34] J. Panero and M. Zelnik, *Human Dimension and Interior Space: A Source Book of Design Reference Standards*. New York: Watson-Guptill, 1979.

[35] D. Reisfeld, H. Wolfson, and Y. Yeshurun, "Context free attentional operators: The generalized symmetry transform," *Int. J. Comput. Vis.*, vol. 14, no. 2, pp. 119–130, Mar. 1995.

[36] N. Robertson and I. Reid, "Estimating gaze direction from low-resolution faces in video," in *Proc. ECCV*, 2006, pp. 402–415.

[37] D. Russakoff and M. Herman, "Head tracking using stereo," *Mach. Vis. Appl.*, vol. 13, no. 3, pp. 164–173, 2002.

[38] K. Smith, S. O. Ba, J.-M. Odobez, and D. Gatica-Perez, "Tracking the visual focus of attention for a varying number of wandering people," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, no. 7, pp. 1212–1229, Jul. 2008.

[39] R. Stiefelhagen and J. Zhu, "Head orientation and gaze direction in meetings," in *Proc. Conf. Hum. Factors Comput. Syst.*, 2002, pp. 858–859.

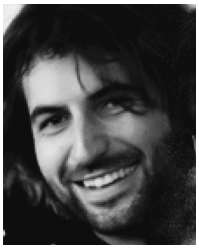
[40] J. Sung, T. Kanade, and D. Kim, "Pose robust face tracking by combining active appearance models and cylinder head models," *Int. J. Comput. Vis.*, vol. 80, no. 2, pp. 260–274, Nov. 2008.

[41] J. Tu, T. Huang, and H. Tao, "Accurate head pose tracking in low resolution video," in *Proc. Autom. Face Gesture Recog.*, 2006, pp. 573–578.

[42] M. Türkan, M. Pardás, and A. Çetin, "Human eye localization using edge projection," in *Proc. Comput. Vis. Theory Appl.*, 2007, pp. 410–415.

[43] R. Valenti, UvAEyes, eye annotations for the Boston University head pose dataset [Online]. Available: http://staff.science.uva.nl/rvalenti/index.php?content=UvAEyes

[44] R. Valenti and T. Gevers, "Accurate eye center location and tracking using isophote curvature," in *Proc. CVPR*, 2008, pp. 1–8.
[45] R. Valenti, J. Staiano, N. Sebe, and T. Gevers, "Webcam-based visual gaze estimation," in *Proc. Int. Conf. Image Anal. Process.*, 2009, pp. 662–671.
[46] R. Valenti, Z. Yucel, and T. Gevers, "Robustifying eye center localization by head pose cues," in *Proc. CVPR*, 2009, pp. 612–618.
[47] P. Viola and M. J. Jones, "Robust real-time face detection," *Int. J. Comput. Vis.*, vol. 57, no. 2, pp. 137–154, May 2004.
[48] M. Voit and R. Stiefelhagen, "Deducing the visual focus of attention from head pose estimation in dynamic multi-view meeting scenarios," in *Proc. ICMI*, 2008, pp. 173–180.
[49] J. Xiao, T. Kanade, and J. Cohn, "Robust full motion recovery of head by dynamic templates and re-registration techniques," in *Proc. FG*, 2002, pp. 163–169.
[50] Y. Zhang and C. Kambhamettu, "3D head tracking under partial occlusion," *Pattern Recognit.*, vol. 35, no. 7, pp. 1545–1557, Jul. 2002.
[51] Z. H. Zhou and X. Geng, "Projection functions for eye detection," *Pattern Recognit.*, vol. 37, no. 5, pp. 1049–1056, May 2004.



Roberto Valenti (S'08–M'11) received the M.Sc. degree with high honors from the University of Amsterdam, Amsterdam, The Netherlands, where he is currently working toward the Ph.D. degree in the Intelligent Systems Laboratorium Amsterdam.

He is a Cofounder and the Chief Technical Officer of ThirdSight, a spinoff of the University of Amsterdam, focused on the automatic analysis of faces. His research mainly focuses on sensing and understanding users' interactive actions and intentions, multimodal and affective human–computer interaction, estimation of the human visual gaze, and behavior analysis.

Nicu Sebe (M'01) is with the Faculty of Cognitive Sciences, University of Trento, Trento, Italy, where he is leading the research in the areas of multimedia information retrieval and human–computer interaction in computer vision applications He has been a Visiting Professor with Beckman Institute, University of Illinois at Urbana-Champaign, Urbana, and with the Department of Electrical Engineering, Darmstadt University of Technology, Darmstadt, Germany.

Prof. Sebe was involved in the organization of the major conferences and workshops addressing the computer vision and human-centered aspects of multimedia information retrieval, among which as a General Cochair of the IEEE Automatic Face and Gesture Recognition Conference (FG 2008), Association for Computing Machinery (ACM) International Conference on Image and Video Retrieval 2007 and 2010, and the Workshop on Image Analysis for Multimedia Interactive Services 2009, and as one of the initiators and a Program Cochair of the Human-Centered Multimedia track of the ACM Multimedia 2007 conference. He is the General Chair of the ACM Multimedia 2013 and a Program Chair of the ACM Multimedia 2011. He has served as the guest editor for several special issues in the IEEE Computer, Computer Vision, and Image Understanding; Image and Vision Computing; Multimedia Systems; and the ACM Transactions on Multimedia Computing, Communications, and Applications. He is the Cochair of the IEEE Computer Society Task Force on Human-Centered Computing and is an Associate Editor of the Machine Vision and Applications, the Image and Vision Computing, the Electronic Imaging, and the Journal of Multimedia.

Theo Gevers (M'01) is an Associate Professor of computer science with the University of Amsterdam (UvA), Amsterdam, The Netherlands, and a Full Professor with the Computer Vision Center, Universitat Autònoma de Barcelona, Barcelona, Spain. With the UvA, he is a Teaching Director of the M.Sc. degree in artificial intelligence. He is the Chair for various conferences and is an Associate Editor for the IEEE TRANSACTIONS ON IMAGE PROCESSING. He is a Cofounder and the Chief Scientific Officer of ThirdSight, a spinoff of the UvA. Furthermore, he is a Program Committee Member for a number of conferences and an invited speaker at major conferences. He is a Lecturer delivering postdoctoral courses given at various major conferences (the IEEE Conference on Computer Vision and Pattern Recognition; the International Conference on Pattern Recognition; SPIE; and the Computer Graphics, Imaging, and Visualization). His main research interests are in the fundamentals of image understanding, object recognition, and color in computer vision. Furthermore, he is interested in different aspects of human behavior, specifically in emotion recognition.

Prof. Gevers currently holds a VICI Award (for research excellence) from the Dutch Organization for Scientific Research.