What are you looking at? Automatic estimation and inference of gaze
Valenti, R.

Citation for published version (APA):

General rights
It is not permitted to download or to forward/distribute the text or part of it without the consent of the author(s) and/or copyright holder(s), other than for strictly personal, individual use, unless the work is under an open content license (like Creative Commons).

Disclaimer/Complaints regulations
If you believe that digital publication of certain material infringes any of your rights or (privacy) interests, please let the Library know, stating your reasons. In case of a legitimate complaint, the Library will make the material inaccessible and/or remove it from the website. Please Ask the Library: http://uba.uva.nl/en/contact, or a letter to: Library of the University of Amsterdam, Secretariat, Singel 425, 1012 WP Amsterdam, The Netherlands. You will be contacted as soon as possible.
Synergetic Eye Center Location and Head Pose Estimation

3.1 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], to analyze the user attention while driving [28], and other applications. It is known that gaze is a product of two contributing factors [66]: the head pose and the eye locations. The estimation of these two factors is often achieved using expensive, bulky or limiting hardware [19]. 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 [69, 90].

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:

1R. Valenti, N. Sebe, and T. Gevers, "Combining Head Pose and Eye Location Information for Gaze Estimation", IEEE Transactions on Image Processing, 2011. Ideas previously appeared in:
R. Valenti, A. Lablack, N. Sebe, C. Djeraba, and T. Gevers, "Visual Gaze Estimation by Joint Head and Eye Information", International Conference on Pattern Recognition, 2010, and
The eye location algorithms found in commercially available eye trackers share the problem of sensitivity to head pose variations, require the user to be either equipped with a head mounted device, or to use a high resolution camera combined with a chinrest 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 which make use of standard low resolution cameras are considered to be less invasive and so more desirable in a large range of applications. Within the appearance-based methods for eye location proposed in literature, [84, 57, 62, 109] 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 variety of applications.

Head pose estimation often requires multiple cameras, or complex face models which requires accurate initialization. Ba et al. [5] 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 fusion of four camera views in [6]. Huang et al. 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 [50]. Hu et al. describe a coarse to fine pose estimation method by combining facial appearance asymmetry and 3D head model [48]. A generic 3D face model and an ellipsoidal head model are utilized in [105] and [2], respectively. In [79] 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 of being able to estimate gaze starting from eye location or head pose only is valid in a limited number of scenarios [96, 115]. 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, while an head pose gaze estimator alone would fail in finely control the cursor on a computer screen.

Hence, a number of studies focused on combining head and eye information for gaze estimation are available in literature: Newman and Matsumoto [83, 75] consider a tracking scenario equipped with stereo cameras and employ 2D fea-
ture tracking and 3D model fitting. The work proposed by Ji et al. [54] 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 to monocular low resolution implementations.

However, no study is performed on the feasibility of an accurate appearance-only gaze estimator which considers both the head pose and the 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 to study 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 orientation of the eyes while the latter 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 posed 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 are the following:

- Rather than just a sequential combination, we propose a unified framework which provides a deep integration of the used head pose tracker and the eye location estimation methods.

- The normal working range of the used eye locator (∼30°) is extended. The shortcomings of the reported eye locators due to extreme head poses are compensated using the feedback from the head tracker.

- 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.

- 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.

The chapter 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 3.2. In Section 3.3, 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 3.4 will describe how the improved estimations could be used to create a combined gaze estimation system. In Section 3.5, three independent experiments will analyze the improvements obtained on the head pose, on the eye location and on the combined gaze estimation. Finally, the discussions and conclusions will be given in Section 3.6.

3.2 Eye Location and Head Pose Estimation

To describe how the used eye locator and head pose estimator are combined in Section 3.3, in this section the used eye locator and the head pose estimator are discussed.

3.2.1 Eye Center Localization

As we are discussing appearance based methods, an overview of the state of the art on the subject is given. The method used by Asteriadis et al. [4] 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. [25] use a multistage approach to detect facial features (among them the eye centers) using a face detector, Pairwise Reinforcement of Feature Responses (PRFR), and a final refinement by using Active Appearance Model (AAM) [22]. Türkan et al. [106] use edge projection (GPF) [122] and support vector machines (SVM) to classify estimates of eye centers. Bai et al. [7] use an enhanced version of Reisfeld’s generalized symmetry transform [88]) for the task of eye location. Hamouz et al. [43] search for ten features using Gabor filters, use features triplets to generate face hypothesis, register them for affine transformations and verify the remaining configurations using two SVM classifiers. Finally, Campadelli et al. [16] 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 SVM on optimally selected Haar wavelet coefficients. With re-
spect to the aforementioned methods, the method proposed in [109] achieves the best results for accurate eye center localization, without heavy constraints on illumination, rotation, and robust to slight pose changes, and will therefore be used.

The method uses isophotes (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:

$$
k = \frac{\delta I^2 \delta^2 I - 2 \frac{\delta I}{\delta x} \frac{\delta^2 I}{\delta x \delta y} \delta I + \frac{\delta I^2 \delta^2 I}{\delta x^2 \delta y^2}}{(\frac{\delta I}{\delta x} + \frac{\delta I}{\delta y})^2}.
$$

where, for example, $\frac{\delta I}{\delta x}$ 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 above term. The orientation is calculated using the gradient but its direction indicates always the highest change in luminance (Figure 3.1(a)). The gradient is then multiplied by 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, $D(x, y)$ are found to be

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

In this way, every pixel in the image gives a rough estimate of its own center the center as shown in Figure 3.1(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 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 non-eye regions or highlights. In order to collect this information and deduce the location of a global eye center, $D(x, y)$'s are mapped into an accumulator (Figure 3.1(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 which follow the edges of the object are considered. This weighting is performed by using curvedness [59]:

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

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 Figure 3.2. For this case, the eye center estimate can clearly be seen on the 3D plot.

In [109], it is shown that the described method yields low computational cost allowing real-time processing. Further, due to the use of isophotes, the method is shown to be robust against linear illumination changes and to moderate changes in head pose. However, the accuracy of the eye center location drops significantly in the presence of head poses which are far from frontal. This is due to the fact that, in these cases, the analyzed eye structure is not symmetric anymore 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 head pose, a head pose estimation algorithm will be employed.

### 3.2.2 Head Pose Estimation

Throughout the years, different methods for head pose estimation have been developed. The 3D model based approaches achieve robust performance and can deal with large rotations. However, most of the method work reasonably in restricted domains only, e.g. some systems only work when there is stereo-data available [78, 92], when there is no (self-) occlusion, or when the head is rotating not more than a certain degree [18]. Systems that solve most of these problems, usually do not work in real-time due to the complex face models they use [120], or require accurate initialization. However, if the face model complexity is reduced to 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 [13, 18, 119]. Among them, the implementation of Xiao et al. [119] works remarkably well. This cylindrical approach is still capable of tracking the head also in situations where the head turns more than 30° from the frontal position and will therefore be used in this work, and will be 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 cylindrical head model 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 the pose $p = [\omega_x, \omega_y, \omega_z, t_x, t_y, t_z]$ according to anthropometric values [41, 27], where $\omega_x$, $\omega_y$, and $\omega_z$ are the rotation parameters and $t_x, t_y, t_z$ are the translation parameters. The eye locations are detected in the face region and are used to give a better estimate of the $t_x$ and $t_y$. The 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 initial pitch ($\omega_x$) and yaw ($\omega_y$) angles are assumed to be null, while the roll angle ($\omega_z$) is initialized by the relative position of the eyes.

To analyze the effect of the motion of the cylindrical head model on the image
frame, the relation between the 3D locations of the points on the cylinder and their corresponding projections on the 2D image plane need to be established. Therefore the 3D 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 3D elliptic cylindrical model, perspective projection is applied to get the corresponding coordinates on the 2D image plane.

Since the cylindrical head model is assumed to be aligned along y-axis of the reference frame and to be positioned such that the center coincides with the origin (as shown in Figure 3.3), any point \( p = (p_x, p_y, p_z)^T \) on the cylinder satisfies the following explicit equation:

\[
\left( \frac{p_x}{r_x} \right)^2 + \left( \frac{p_z}{r_z} \right)^2 = 1, \tag{3.1}
\]

where \( r_x \) and \( r_z \) stand for the radii of the ellipse along x- and z-axes respectively. To calculate the coordinates of the points on the visible part of the cylinder, the front region is sampled in an \( N_s \times N_s \) grid-like structure on \( x-y \) plane and corresponding depth values are obtained by using Equation 3.1. These sampled points are considered to summarize the motion of the cylinder and they are employed in Lukas Kanade optical flow algorithm. The perspective projection of the 3D points on the elliptic cylindrical face model gives the 2D pixel coordinates in the image plane. Let point \( p = (p_x, p_y, p_z)^T \) be a point sampled on the cylinder and point \( u = (u_x, u_y)^T \) be its projection on the image plane. Figure 3.4 illustrates the side view of this setting by making a pin hole camera assumption for the sake simplification. Using similarity of triangles in Figure 3.4, the following equations apply for the relation between \( p \) and \( u \)
3.2 Eye Location and Head Pose Estimation

Figure 3.4: Perspective projection of point $p$ onto image plane by a pin hole camera assumption

$$
p_x = \frac{p_x u_x}{fl},
$$
$$
p_y = \frac{p_y u_y}{fl},
$$

(3.2)

where $fl$ stands for the focal length of the camera. This relation is summarized by a perspective projection function $P$, which maps the 3D points onto the 2D image plane employing the above given identities,

$$
P(p) = u.
$$

As shown in Figure 3.3, 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 pose vector $p_i$ by the rigid motion vector $\Delta \mu_i = [\omega_i^x, \omega_i^y, \omega_i^z, \tau_i^x, \tau_i^y, \tau_i^z]$. 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 formulation of this relation, three transformation functions are employed as illustrated in Figure 3.3. The 3D transformation function $M$ maps $p_i$ to $p_{i+1}$, whereas the 2D transformation function $F$ maps $u_i$ to $u_{i+1}$ and the perspective projection function $P$ 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 [119]:

$$
P(M(p_i, \Delta \mu_i)) = \frac{p_i^{x} \omega_z + p_i^{y} \omega_x + p_i^{z} \omega_y + \tau_x}{p_i^{x} \omega_y + p_i^{y} \omega_z + p_i^{z} + \tau_y} \\ \times \frac{p_i^{y} \omega_x + p_i^{z} \omega_y + \tau_y}{p_i^{x} \omega_y + p_i^{y} \omega_z + p_i^{z} + \tau_y}
$$

In the next section, the estimated head pose will be used to obtain the pose normalized eye patches.
3.3 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 work adequately under certain circumstances. In [109], 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 head pose, the system can still be 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 be able to recover the correct track. By integrating the eye locator with the cylindrical head model, we aim to obviate these drawbacks.

Instead of a sequential integration of the two systems, an early integration is proposed. Relevant to our work is the approach proposed in [100]. The authors combine a cylindrical head model with an Active Appearance Model (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 cylindrical head model together with the eye locator proposed in [109] to broaden the capabilities of both systems and to improve the accuracy of each individual component. By comparing the transformation matrices suggested independently 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 3D space, the 2D eye locations
detected in the first frame are used as reference points (e.g. the "+" markers in Figure 3.7). These reference points are projected onto the cylindrical head model, 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.

3.3.1 Eye Location by Pose Cues

![Figure 3.6: 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.](image)

Around each reference point projected onto the 3D model, an area is sampled and transformed by using the transformation matrix obtained by the head pose tracker (Figure 3.5). The pixels under these sampled points are then remapped into a normalized canonical view (Figure 3.6). Note that extreme head poses are also successfully corrected, although some perspective projection errors are retained. The eye locator described in Section 3.2.1 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 estimated eye center (the white dots in Figure 3.6 and the "x" markers in Figure 3.7). In this way, as long as the CHM tracker is correctly estimating the head pose, the localized eyes can
be considered to be optimal. Figure 3.7 shows two examples in which the default eye locator would fail ("." marker) but the pose normalized eye estimation would be correct ("x" marker).

### 3.3.2 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 3D 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 [79], 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, while 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 Figure 3.8, while its pseudo code is presented in Algorithm 2.
3.3 Synergetic Eye Location and CHM Tracking

Algorithm 2 Pseudo-code 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 distance between the eyes to get the depth element, \(t_z\).
- Initialize pose \(p\) using eye locations

**Iterate through all the frames**
for \(t = 0\) to last frame number do
  - Assume intensity is constant between consecutive frames, \(I_{t+1} = I_t\).
  - Compute the gradient \(\nabla I_{t+1}\) 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 \(m = p_{t+1} \ast p_t^{-1}\)
  - Load Gaussian pyramid image \(I(l)\)
  - Initialize \(\Delta \vec{p} = [0, 0, 0, 0]\)
  while maximum iterations not reached or \(\Delta \vec{p} <\) threshold do
    - Transform pixels \(p\) of \(I(l)\) to \(p'\) with transformation matrix \(M\) and parameters \(p\) to compute \(I_t(p)\)
    - 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, \(p_x = \vec{u}_n \ast t_z, p_y = \vec{v}_n \ast t_z\)
    - With back-projection calculate pixels \((u', v')\)
    - Compute \(\nabla I_{t+1}(m) \frac{\partial T}{\partial p}\) where \(T\) summarizes the projection model.
    - Compute Hessian matrix in
    - Compute \(\Delta \vec{p}\) using
    - Update the pose and motion:
      - \(p_{t+1} = \Delta \vec{p} \circ p_t + p_{t+1}\)
      - \(m = \Delta \vec{p} \circ m\)
  end while
  - Update transformation matrix \(M = \Delta \vec{p} \circ M\)
  - Transform reference eye regions \(R_r\) and \(R_l\) using \(M\)
  - Remap eye regions to pose normalized view
  - Compute displacements vectors \(D\) on pose normalized eye regions accordingly to [109], using,
    \[
    D(x, y) = -\frac{\partial I_t}{\partial x} \frac{\partial^2 I_{t+1}}{\partial x^2} + 2 \frac{\partial I_t}{\partial x} \frac{\partial I_{t+1}}{\partial x} \frac{\partial^2 I_{t+1}}{\partial x \partial y} + \frac{\partial^2 I_t}{\partial y^2} \frac{\partial^2 I_{t+1}}{\partial x \partial y} + \frac{\partial^2 I_t}{\partial y^2} \frac{\partial^2 I_{t+1}}{\partial y^2}
    \]
  - Vote for centers weighted by \(\frac{\partial I_t}{\partial x} \frac{\partial^2 I_{t+1}}{\partial x^2} + 2 \frac{\partial I_t}{\partial x} \frac{\partial I_{t+1}}{\partial x} \frac{\partial^2 I_{t+1}}{\partial x \partial y} + \frac{\partial^2 I_t}{\partial y^2} \frac{\partial^2 I_{t+1}}{\partial x \partial y} + \frac{\partial^2 I_t}{\partial y^2} \frac{\partial^2 I_{t+1}}{\partial y^2}\)
  - Select isocenter closer to the center of eye region as eye estimate
  - Remap eye estimate to cylinder coordinates
  - Create pose vector from eye location and compare it to head tracker's
    if distance between pose vector > threshold then
      average pose vectors and create the new \(M\)
    end if
end for
3.4 Visual Gaze Estimation

In the previous section, we described how the 2D eye center locations detected in the first frame are used as reference points (Figure 3.12). These reference points are projected onto the cylindrical head model and are then used to estimate the successive, pose normalized eye center locations. In this section, 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.

3.4.1 The Human Visual Field of View

Studies on the human visual field of view [85] show that, while looking straight ahead, it has a vertical span of 130° (60° above and 70° below) and approxi-
mately 90° on each side, which corresponds to a photographic objective angle of 180° (Figure 3.9). The common field of view of the two eyes is called 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 a pyramid $OABCD$ where $O$ represents the point between the two eyes and rectangle $ABCD$ represents the visual field of view at distance $d$. Further, the angles $\alpha$ and $\beta$ denote the horizontal and vertical angles of the visual field of view in binocular human vision, respectively [63]. Since the pyramid is an approximation of the field of view, we are able to center it on the gaze point $P$ so that it is in the middle of the field of view. In this case, the vector $OP$ denotes the visual gaze vector (Figure 3.10).

The width ($W$) and height ($H$) of the visual field at distance $d$ are computed by:

$$W = 2PF = 2d \tan \frac{\alpha}{2}, \quad H = 2PG = 2d \tan \frac{\beta}{2}.$$

The projection of the visual field of view on the gazed scene in front of a user is a quadrilateral $A'B'C'D'$ with central gaze point $P'$, 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 $(OP)$. The head pose parameters computed by the method described in Section 3.2.2 are used to define the projection of the region of interest in the target scene.

### 3.4.2 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 plane [111]. In this way, the problem of eye displacement is simply solved by a 2D 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 infrared anchor points. However, this approach forces the user to use a chin rest to avoid head movements which will result in wrong mappings.

In this work, 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 of [99], where it is shown that head
3.4 Visual Gaze Estimation

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 3D gaze estimation is reduced to the sub-problem of estimating it in 2D (e.g. using eyes only), removing the constraints on head movements.

Instead of learning all possible calibrations in 3D space, we propose to automatically retarget a set of known points on a target plane (e.g. a computer screen) in order to simulate a re-calibration each time the user moves his/her head. In fact, if the known points are translated accordingly to the parameters obtained from the head pose, it is possible to use the previously obtained displacement vectors and re-calibrate 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 Figure 3.11 (a), so that it can be moved using the same transformation matrix obtained from the head pose estimator (to ensure that it moves accordingly). 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 3D 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. Figure 3.11(b) shows how the calibration points are projected on the calibration plane and Figure 3.11 (c) illustrates how these points change during head movements, obtaining new intersections on the target plane (the Pose-Retargeted known points).
3.5 Experiments

Here, three components need an independent evaluation: (1) the accuracy provided by the eye center location given the head pose, (2) the accuracy obtained by the head pose estimation given the eye center location and (3) the accuracy of the combined final visual gaze estimation. In the following sections, the datasets, error measures, and the result for each of three components are described and discussed.

3.5.1 Eye Location Estimation

The performance obtained by using head pose cues in eye location are evaluated using the Boston University head pose database [17]. The database consists of 45 video sequences, where 5 subjects were asked to perform 9 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 degrees in rotation. However, Cascia et al. [17] 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 ground truth. As no annotation of the eye location on this dataset is available, we manually annotated the eyes of the subjects on 9000 frames. These annotations are publicly available at [108].

In quantifying the error, we used the 2D normalized error. This measure was introduced by Jesorsky et al. [52] and is widely used in eye location literature [7, 16, 43, 106, 122]. The normalized error represents the error obtained by the worse eye estimation and is defined as:

\[ e = \max(d_{\text{left}}, d_{\text{right}}) / d \]  

where \( d_{\text{left}} \) and \( d_{\text{right}} \) 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) corresponds roughly to the distance between the eye center and the
Figure 3.12: A comparison between the eye detection results with and without pose cues

Table 3.1: Effect of pose cues in eye localization

<table>
<thead>
<tr>
<th></th>
<th>Worse eye</th>
<th>Best eye</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Without pose</td>
<td>With pose</td>
</tr>
<tr>
<td>$e \leq 0.05$</td>
<td>40.6</td>
<td>31.93</td>
</tr>
<tr>
<td>$e \leq 0.1$</td>
<td>61.73</td>
<td>77.27</td>
</tr>
<tr>
<td>$e \leq 0.15$</td>
<td>66.14</td>
<td>88.46</td>
</tr>
<tr>
<td>$e \leq 0.2$</td>
<td>70</td>
<td>93.67</td>
</tr>
<tr>
<td>$e \leq 0.25$</td>
<td>77.72</td>
<td>96.74</td>
</tr>
</tbody>
</table>

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 Figure 3.12 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 Figure 3.12 together with the baseline accuracy obtained by the standard eye locator [109]. In the latter, the approximate face position is estimated using the boosted cascade face detector proposed by Viola and Jones [114], where the rough positions of the left and right eye regions are estimated by anthropometric relations [41]. 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
dataset. The working range of the face detector is around 30° around each axis, while certain head poses in the dataset 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 3.1 for several normalized error values.

From Figure 3.12, 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 Figure 3.6, 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.

### 3.5.2 Head Pose Estimation

**Table 3.2: Comparison of RMSE and STD**

<table>
<thead>
<tr>
<th></th>
<th>Sung et al. [100]</th>
<th>An et al. [2]</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>RMSE</td>
<td>STD</td>
</tr>
<tr>
<td><strong>Pitch ($\omega_x$)</strong></td>
<td>5.26</td>
<td>4.67</td>
</tr>
<tr>
<td><strong>Yaw ($\omega_y$)</strong></td>
<td>6.10</td>
<td>5.79</td>
</tr>
<tr>
<td><strong>Roll ($\omega_z$)</strong></td>
<td>3.00</td>
<td>2.82</td>
</tr>
</tbody>
</table>

Since the ground truth is provided by the Boston University head pose database [17], 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: $\omega_x$, $\omega_y$, and $\omega_z$.

To measure the accuracy of 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 3.2 shows the improvement in RMSE and 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 by 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 estimation of the pose angles. Some challenging examples of the results obtained by our implementation of the CHM head pose tracker are represented in Figure 3.13 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 Figure 3.14. It can be derived that the system is able to cope with these extreme head poses.

In the last two columns of Table 3.2, we compare our results with two other methods in the literature, which use the same database. Similar to our method, Sung et al. [100] propose a hybrid approach combining active appearance models and cylinder head models to extend the operating range of AAM. An et al. [2] propose to replace the traditional CHM with a simple 3D ellipsoidal model. They provide comparison of accuracy with planar and cylindrical models. Here, we consider the accuracy reported by Sung et al. and from An et al. on the cylindrical head model [2]. From Table 3.2, 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 by 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 which 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 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 semi-frontal face is detected again.

### 3.5.3 Visual Gaze Estimation

This section describes the experiments performed to evaluate the proposed gaze estimation system. To this end, a heterogeneous dataset was collected, which
includes 11 male and female subjects with different ethnicity, with and without glasses and different illumination conditions. Figure 3.15 shows some examples of the subjects in the dataset. The data was 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 × 576 pixels and the resolution of the computer screen is 1280 × 1024 pixels. To test the system under natural and extreme head movements, the subjects were requested to perform two set of experiments:

![Figure 3.15: Some of the test subjects](image)

(1) 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 Figure 3.16(a)) and move their head around while still looking at the specific point. The point is displayed at certain locations on the screen for about 4 seconds 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 towards 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 is shown in Figure 3.16(a). (2) 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 Figure 3.16(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 meth-
Eyes Only Gaze Estimator: This estimator uses the anchor-pupil vectors directly into the mapping as in the system proposed in [111]. 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 presence of head movements.

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 cylindrical head model. 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 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;

Pose-Retargeted Gaze Estimator: This is the approach proposed in Section 3.4.2, which treats the 3D gaze estimation problem as a superset of 2D problems. Also 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 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 3.3 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
3.5 Experiments

<table>
<thead>
<tr>
<th>Subject</th>
<th>Eyes Only</th>
<th>Pose-Normalized</th>
<th>Pose-Retargeted</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Mean</td>
<td>Std</td>
<td>Mean</td>
</tr>
<tr>
<td>1</td>
<td>688.91</td>
<td>281.29</td>
<td>1090.71</td>
</tr>
<tr>
<td>2</td>
<td>633.08</td>
<td>187.74</td>
<td>464.25</td>
</tr>
<tr>
<td>3</td>
<td>2285.11</td>
<td>436.26</td>
<td>4929.71</td>
</tr>
<tr>
<td>4</td>
<td>1202.11</td>
<td>2664.01</td>
<td>2537.21</td>
</tr>
<tr>
<td>5</td>
<td>1388.72</td>
<td>276.79</td>
<td>1073.91</td>
</tr>
<tr>
<td>6</td>
<td>710.24</td>
<td>328.63</td>
<td>443.08</td>
</tr>
<tr>
<td>7</td>
<td>5082.26</td>
<td>731.92</td>
<td>19725.89</td>
</tr>
<tr>
<td>8</td>
<td>8482.27</td>
<td>693.31</td>
<td>14280.32</td>
</tr>
</tbody>
</table>

Table 3.3: Mean pixel error and standard deviation comparison on the static dot gazing task

head pose while keeping the eyes fixed, the Eyes Only estimator has a significantly larger error and standard deviation 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 \((8.5^\circ, 6.8^\circ)\), while the Pose-Retargeted estimator has a mean error of \((210.33, 214.99)\) pixels, corresponding to an angle of \((4.6^\circ, 4.7^\circ)\) 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 compared to the Pose-Normalized system.

<table>
<thead>
<tr>
<th>Subject</th>
<th>Eyes Only</th>
<th>Pose-Normalized</th>
<th>Pose-Retargeted</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Mean</td>
<td>Std</td>
<td>Mean</td>
</tr>
<tr>
<td>1</td>
<td>3461.68</td>
<td>935.38</td>
<td>1931.83</td>
</tr>
<tr>
<td>2</td>
<td>3125.42</td>
<td>361.55</td>
<td>5874.41</td>
</tr>
<tr>
<td>3</td>
<td>3531.19</td>
<td>564.11</td>
<td>7725.46</td>
</tr>
<tr>
<td>4</td>
<td>2380.21</td>
<td>401.89</td>
<td>2002.35</td>
</tr>
<tr>
<td>5</td>
<td>3554.94</td>
<td>665.56</td>
<td>2799.42</td>
</tr>
<tr>
<td>6</td>
<td>2658.84</td>
<td>472.37</td>
<td>1574.86</td>
</tr>
<tr>
<td>7</td>
<td>3606.85</td>
<td>729.86</td>
<td>3414.06</td>
</tr>
<tr>
<td>8</td>
<td>3332.96</td>
<td>573.66</td>
<td>6989.07</td>
</tr>
<tr>
<td>9</td>
<td>11958.67</td>
<td>775.88</td>
<td>21508.32</td>
</tr>
<tr>
<td>10</td>
<td>89.36</td>
<td>109.38</td>
<td>72.27</td>
</tr>
</tbody>
</table>

Table 3.4: Mean pixel error and standard deviation comparison on the dot following task

Table 3.4 shows the results of the second task (dot following). In this task, due to the fact that the head significantly shifts 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 $(5.8^\circ, 3.0^\circ)$, while the Pose-Retargeted estimator has a mean error of $(87.18, 103.86)$ pixels, corresponding to an angle of $(1.9^\circ, 2.2^\circ)$ in the $x$ and $y$ direction, respectively. When compared to 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 Figure 3.17.

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 of the results obtained by the proposed system. By analyzing the causes for the errors (and the big standard deviation) 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 so 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 $x$ direction. Furthermore, errors in the eye center locator seriously affect the system, as just a few pixels error 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, while the Pose-Normalized estimator clearly outperforms the method based on 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.

### 3.6 Conclusions

In this chapter, we proposed a deep integration of a CHM based head pose tracker and an isophote based eye locator in a complementary manner, so that both system can benefit from each other’s evidence. Experimental results showed that the accuracy of both independent systems is improved by their combination. The eye location estimation of the unified scheme achieved an improve-
3.6 Conclusions

Figure 3.17: Errors for the three tested estimators compared to the computer screen for the (a) static dot gazing task and (b) dot following task

ment in accuracy from 16% to 23%, while the pose error has been improved from 12% to 24%. Besides the improvements in accuracy, the operating range of the eye locator has been extended (more than 15°) by the head tracker and the ineffectiveness of the previously reported eye location methods against extreme head poses was compensated. Furthermore, automatic quality control and re-initialization of the head tracker was 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 is 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 dataset proved that 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.