Int J Comput Vis (2013) 101:459–481
DOI 10.1007/s11263-012-0541-8

Improving Head Movement Tolerance of Cross-Ratio Based Eye Trackers

Flavio L. Coutinho · Carlos H. Morimoto

Received: 15 November 2011 / Accepted: 19 May 2012 / Published online: 6 June 2012

Abstract When first introduced, the cross-ratio (**CR**) based remote eye tracking method offered many attractive features for natural human gaze-based interaction, such as simple camera setup, no user calibration, and invariance to head motion. However, due to many simplification assumptions, current **CR**-based methods are still sensitive to head movements. In this paper, we revisit the **CR**-based method and introduce two new extensions to improve the robustness of the method to head motion. The first method dynamically compensates for scale changes in the corneal reflection pattern, and the second method estimates true coplanar eye features so that the cross-ratio can be applied. We present real-time implementations of both systems, and compare the performance of these new methods using simulations and user experiments. Our results show a significant improvement in robustness to head motion and, for the user experiments in particular, an average reduction of up to 40 % in gaze estimation error was observed.

Keywords Eye tracking · Gaze tracking · Remote eye gaze tracking · Head movement tolerance · Free-head motion · Cross-ratio · Homography

1 Introduction

Natural human interfaces can benefit from eye movement information (Duchowski 2003). Several methods have been developed to track eye movements as described in Morimoto and Mimica (2005); Villanueva et al. (2008); Hansen and Ji (2010). Since we are primarily interested in the use of eye trackers for interactive applications our focus is on devices that are non intrusive and remote. This way, devices that use special contact lenses (Robinson 1963) or electrodes around the eyes (Kaufman et al. 1993) are less interesting, since they require preparation before use and use for long periods of time can be uncomfortable.

Camera based devices overcame these limitations, specially those that use remote configurations, i.e., where the user does not need to wear or to be in contact with any kind of equipment. For natural human interaction, it is also desirable to have remote eye trackers that allow free head movement, which improve usability and comfort.

In general, camera based devices capture and process images of a person's eye. During image processing, relevant eye features are detected and tracked, and used to compute the point of regard (PoR). Typical eye features used are the iris and pupil borders, eye corners, and corneal reflections generated by light sources (active illumination) (Villanueva et al. 2008).

Remote eye gaze tracking methods can be classified into two groups (Hansen and Ji 2010): interpolation based methods and model based. Interpolation based methods map image features to gaze points. Model based methods estimate the 3D gaze direction and intersection between scene geometry and gaze direction is computed as the PoR. System requirements of interpolation based methods tend to be smaller than model based methods but head movement is restricted. Model based methods, on the other hand, offers greater freedom of movement though they require more complex system setup.

Cross-ratio (**CR**) based methods combine advantages from both interpolation and model based methods: they

F.L. Coutinho (✉) · C.H. Morimoto
Department of Computer Science, University of São Paulo, São Paulo, Brazil
e-mail: flc@ime.usp.br

C.H. Morimoto
e-mail: hitoshi@ime.usp.br

do not require system calibration and they allow free head motion. Unfortunately, due to simplifications assumed by many implementations of the method, the performance of the method might be limited in accuracy or robustness to head movement.

The next section briefly describes common methods for remote eye tracking and revisits the basic solution proposed by **CR** method as well as newer extensions currently presented in the literature. Section 3 discusses some head movement tolerance issues related to current **CR** based methods that motivated the development of this work. Sections 4 and 5 introduce two new methods to improve the robustness of the **CR** method under head motion: the *Cross-ratio with Dynamic Displacement Vector Correction* (**CR-DD**) method and the *Planarization of CR Features* (**PL-CR**) method. Evaluation of the proposed methods by simulation and a user experiment, demonstrating significant improvements, are shown in Sect. 6. A real time implementation of both methods is presented in Sect. 7. Finally, Sect. 8 concludes the paper.

2 Remote Eye Gaze Tracking

As mentioned in the previous section, remote eye gaze tracking methods can be classified into two groups (Hansen and Ji 2010): interpolation based and model based. The *Pupil-Corneal Reflection* method (**PCR**) is an example of an interpolation based gaze tracking technique. The **PCR** method detects and tracks the pupil and a corneal reflection, generated by a light source. Infrared light sources are often used as they do not distract users, offer a more homogeneous lighting condition and improve the robustness to ambient light changes in indoor environments.

Figure 1 illustrates the geometric setup considered by the **PCR** method. Assuming that the cornea surface is a sphere centered at C, the corneal reflection G and its image g do not move when the eye rotates around C. Thus, g can be used as a reference point. As the eye rotates to gaze at different targets, the pupil center P moves in space, and G and P define an image vector **gp** which is mapped to screen coordinates through a mapping function. The mapping function is obtained by a calibration procedure in which the user is asked to gaze at specific screen targets. The work by Morimoto et al. (1999) uses a second order polynomial as a mapping function since a linear mapping may not be adequate for large eye rotations (Cerrolaza et al. 2008).

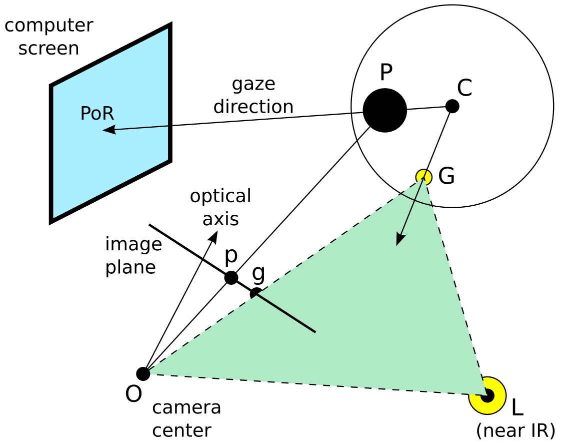


Fig. 1 Geometric setup of the **PCR** method

Since the observed **gp** vector is a function of the scene geometry (camera, eye and screen), different eye positions will define different vectors for the same gaze point. As a consequence, it is not expected that the mapping function, once optimized for the calibration position, will estimate gaze with the same accuracy for different eye positions. This accuracy decay was evaluated by Morimoto and Mimica (2005) and illustrates two limitations of the **PCR** method: low tolerance to head movements and the need of frequent recalibration.

2.1 Model Based Methods

Model based methods use geometric models of the eye to estimate the line of sight in 3D (Shih and Liu 2003; Guestrin and Eizenman 2006, 2008; Hennessey et al. 2006; Chen et al. 2008; Nagamatsu et al. 2008, 2010b; Model and Eizenman 2010). An eye model that is usually considered for model based methods is shown in Fig. 2. Important elements of this model for gaze tracking methods are: the eyeball, modeled as a sphere; the foveola, the central region of the fovea (the retinal region on the back of the eye that is responsible for the detailed vision) that comprehends about 1.3° of visual angle (Duchowski 2003); the pupil, a circular orifice defined by the iris, by which light enters into the eye; the cornea, a transparent membrane that covers the iris and can be approximated by a spherical surface; the optical axis of the eye, the line defined by the centers of the eyeball, cornea, and pupil; and, finally, the visual axis of the eye, the line that

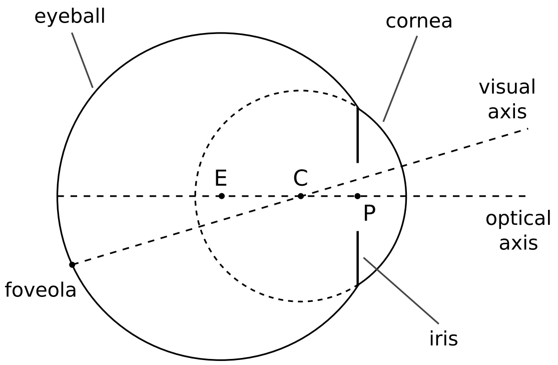


Fig. 2 Geometric eye model and relevant elements for gaze tracking methods

connects the foveola and the point of regard, and also passes through the cornea center (usually pointing in the nasal direction). Average values for this model are: cornea radius of 0.78 cm; distance from the pupil center to cornea center of 0.42 cm; horizontal and vertical angles between the visual and optical axes of 5° and 1.5° (the combined angle between the axes is usually referred to as the κ angle); and a combined index of refraction of 1.3375 for the cornea and aqueous humour (Guestrin and Eizenman 2006).

All model based methods follow a common strategy: first the optical axis of the eye is reconstructed in 3D; the visual axis (deviated from the optical axis by the κ angle) is reconstructed next; finally, the PoR is estimated by intersecting the visual axis with the scene geometry. Reconstruction of the optical axis is done by estimation of the cornea (C) and pupil (P) centers. Since the line of sight is defined by the visual axis and not the optical axis of the eye, the angular deviation between them must be known in order to reconstruct the visual axis from the optical axis.

Some of these model based methods use stereo cameras (Shih and Liu 2003; Guestrin and Eizenman 2008; Chen et al. 2008; Nagamatsu et al. 2008), while a single camera is used by others (Guestrin and Eizenman 2006; Hennessey et al. 2006). In either case, the cameras need to be calibrated and the scene geometry must be known so that the PoR can be computed. Thus, these systems need to be fully calibrated, a requirement that does not exist for the **PCR** technique. This way, freedom of movement is achieved by an increase in the complexity of system setup.

Guestrin and Eizenman (2006) showed that, for gaze estimation methods based on detection and tracking of the pupil and corneal reflections, the complexity of the required eye model varies with the number of cameras and light sources available. The minimum system configuration that allows for free head motion uses a single camera and 2 light sources. With such setup, an eye model with 5 known parameters must be used: cornea radius; distance from pupil center to cornea center; combined index of refraction of the cornea and aqueous humour; and vertical and horizontal components of the κ angle. These personal parameters are estimated by a one time per subject calibration procedure. When a setup that uses at least 2 cameras and at least 2 light sources is used, the optical axis of the eye can be reconstructed without the use of any personal parameters. Horizontal and vertical components of the κ angle still need to be known in order to reconstruct the visual axis in 3D, but the number of calibration points required to estimate these parameters is reduced to 1.

Nagamatsu et al. (2010b) presented a model based method that completely eliminates personal calibration requirements by using a binocular setup (both eyes are tracked simultaneously). They assume that the visual axes of the left and right eyes are symmetric about the sagital plane, and ignore the vertical angle of the visual axis due to its typical low values. By these assumptions, the PoR is computed as the mid point of the points obtained by the intersections of both optical axis with the screen. Since reconstruction of the optical axis for each eye requires the use of 2 cameras, a total of 4 cameras are used by this method.

A similar approach for a user-calibration-free gaze estimation system was proposed by Model and Eizenman (2010). They also use a binocular solution, but do not assume symmetry between the visual axes. Their method estimates horizontal and vertical rotation angles of the visual axis for both eyes (4 parameters in total) during eye tracking usage, but without relying on subjects to stare at specific points on screen. Assuming that at each time instant both visual axes stare at the same point, the 4 parameters are estimated by minimizing the distance between the intersections of both visual axes with the screen.

2.2 Cross-Ratio Based Eye Tracking

A method for remote eye gaze tracking based on the cross-ratio invariant property of projective geometry was introduced by Yoo et al. (2002). The method uses 4 light sources arranged in a rectangular shape, attached over a surface of interest. Typically this surface is the computer screen and each light source is placed at a screen corner. When a person faces the screen with these lights attached, 4 corneal reflections are generated on the cornea surface. These reflections, together with the observed pupil center, are then used to compute the PoR. Figure 3 illustrates the geometric setup considered for this method, where the following elements can be pointed:

- L_i: light sources (screen corners).
- G_i: corneal reflections of L_i.
- g_i: projections of G_i in the image.
- J: point of regard.
- P: pupil center.
- C: center of curvature of the cornea.
- p: image of P.
- O: camera projection center.

In its basic form, the cross-ratio based method for remote eye gaze tracking assumes g_i as projection of G_i, G_i as projection of L_i, and that each one of these groups (g_i, G_i and L_i) is coplanar, defining the planes Π_g, Π_G and Π_L (note that Π_g is coincident with the image plane). Besides that, p (in Π_g) is the projection of P (in Π_G) and P is projection of J (in Π_L).

Points in Π_g result from the composition of two projective transformations over Π_L, and therefore the composition is also a projective transformation. This way, being invariant to this kind of transformation (Hartley and Zisserman 2000),

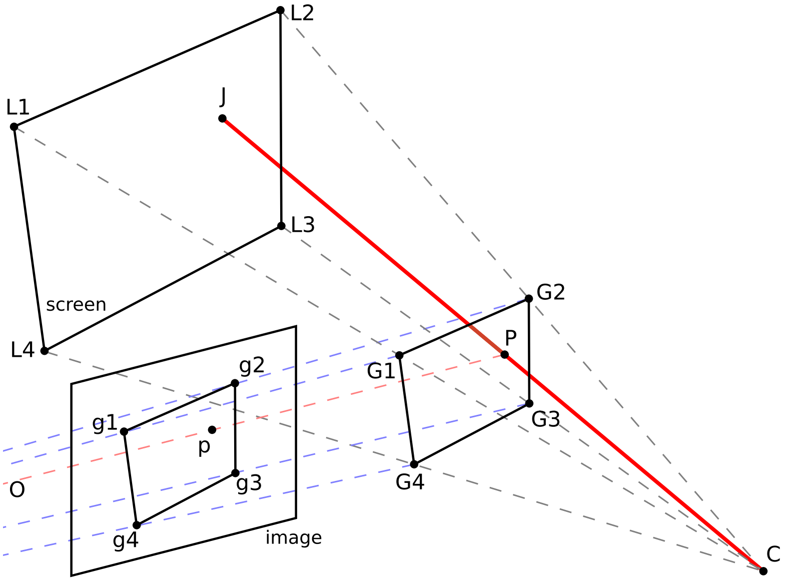


Fig. 3 Geometric setup used by the cross-ratio method for remote eye gaze tracking

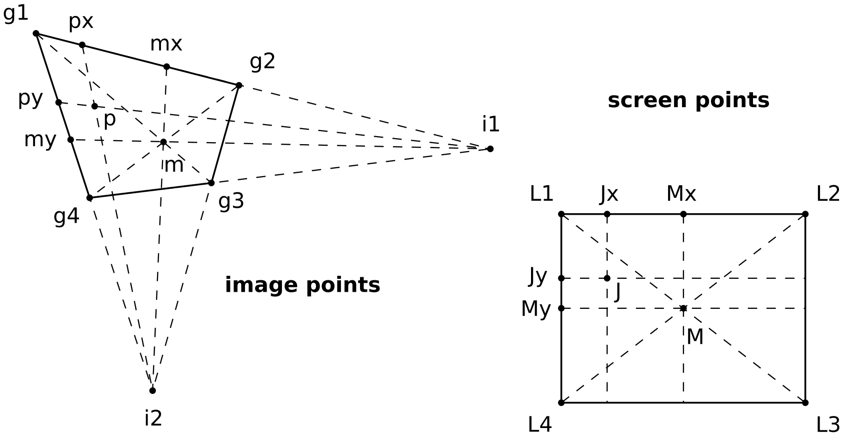


Fig. 4 Estimation of the PoR J using the cross-ratio invariant. First i_1 and i_2 are computed using points g_i. Next, (p_x, p_y) and (m_x, m_y) are computed and used to estimate (J_x, J_y)

the cross-ratio can be used to compute J. The cross-ratio (cr) is defined for 4 collinear points (Q_1, Q_2, Q_3, Q_4) by:

$$cr(Q_1, Q_2; Q_3, Q_4) = \frac{\delta_{13}\delta_{24}}{\delta_{14}\delta_{23}} \tag{1}$$

where δ_{mn} is the Euclidean distance between points Q_m and Q_n.

Figure 4 shows how the cross-ratio invariant can be applied to compute J. From image points g_i, it is possible to compute m (the projection of M, the center point of the rectangle formed by L_i), as well as the ideal points i_1 and i_2. An important property of ideal points is that the images of any pair of parallel lines crosses at their correspondent ideal point. This way, as $\overline{L_1L_2}$ and $\overline{L_3L_4}$ are parallel at the computer screen, $\overline{g_1g_2}$ and $\overline{g_3g_4}$ can be used to compute i_1. By geometric construction, lines $\overline{i_1p}$ and $\overline{i_1m}$ can be used to determine p_y and m_y as shown in Fig. 4. Similarly, p_x and m_x can be determined, being possible to define 2 sets, each one with 4 collinear points: $\{g_1, p_x, m_x, g_2\}$ and $\{g_1, p_y, m_y, g_4\}$. For these 2 sets of points, the following ratios can be computed:

$$r_1 = cr(g_1, p_x, m_x, g_2), \tag{2}$$

$$r_2 = cr(g_1, p_y, m_y, g_4) \tag{3}$$

Due to the cross-ratio invariance to projective transformations, it is also known that:

$$r_1 = cr(L_1, J_x, M_x, L_2), \tag{4}$$

$$r_2 = cr(L_1, J_y, M_y, L_4) \tag{5}$$

Thus, given ratios r_1 and r_2, J_x and J_y (only unknown values in Eqs. (4) and (5)) can be computed, and the PoR J can finally be estimated.

Since the cross-ratio method is based on projective transformations between planes, these transformations can also be described by means of homographies (Hartley and Zisserman 2000). In this case, p can be expressed as

$$p = \mathbf{H_2}\big(\mathbf{H_1}(J)\big) \tag{6}$$

where $\mathbf{H_1}$ is the homography that transforms points from Π_L to Π_G and $\mathbf{H_2}$ the one that transforms points from Π_G to Π_g. Homographies $\mathbf{H_1}$ and $\mathbf{H_2}$ can be combined into a single transformation $\mathbf{H}$ that directly transforms points from Π_L to points in Π_g. Matrix $\mathbf{H}$ can be estimated from the correspondence between points g_i and L_i, and J can be computed as

$$J = \mathbf{H}^{-1}(p) \tag{7}$$

To facilitate the presentation and discussion of other gaze tracking methods based on the cross-ratio concept, we will define the $\mathbf{CR_f}$ function. The $\mathbf{CR_f}$ function receives g_i, p, and the dimensions of the rectangle formed by L_i as input. It returns the point in Π_L that corresponds to p in Π_g. Since the dimensions of the rectangle formed by L_i are usually constant considering a typical gaze tracking scenario, we can drop the dimensions from the input arguments of the $\mathbf{CR_f}$ function. Thus, we will define the following notation for this function:

$$\text{PoR} = \mathbf{CR_f}(g_i, p) \tag{8}$$

For the basic form of the cross-ratio (**CR**) method described until now, gaze estimation procedure can be represented in a compact way by the $\mathbf{CR_f}$ function.

Observe that, in theory, this solution for remote gaze estimation does not impose any restriction on the eye position and no previous parameter value needs to be used. It is, therefore, an elegant and simple solution that tolerates head movements and is calibration-free. Unfortunately, large gaze estimation errors are observed when the **CR** method is used in its basic form. Coutinho and Morimoto (2006), and Guestrin et al. (2008) made detailed investigations to explain the large observed estimation error, identifying two major sources of error which are, in fact, two simplifying assumptions that are not valid in practice. These assumptions are:

1. P and G_i are coplanar.
2. $\overrightarrow{CP}$ is considered as the line of sight.

The first assumption is inaccurate because the location of P relative to Π_G is a function of the distance between P and C, as well as the current eye rotation. Moreover, the location of Π_G is also dependent on L_i, C, O and the cornea radius. Therefore, there is no guarantee that P and Π_G will be coplanar for arbitrary situations. Since the **CR** method is based on transformations between planes, gaze estimation error will increase as the distance from P to Π_G increases. It is also important to note that G_i are, in fact, not coplanar, although an approximation to a plane (Π_G) is reasonable (Hansen et al. 2010).

The second assumption affects gaze estimation results due to the fact that the visual axis of the eye (the actual line of sight) is deviated by κ from the optical axis. When J is computed, what is actually being computed is the point where the optical axis intercepts the screen plane. The point intercepted by the visual axis is displaced from J, and the observed displacement is a function of both eye distance and rotation relative to the screen plane.

When a more realistic geometric setup, as shown in Fig. 5, is considered, the **CR** method cannot be directly applied to estimate the PoR as shown in Eq. (8). Consider the following elements of this new setup:

- L_i: light sources (screen corners).
- G_i: corneal reflections of L_i.
- g_i: projection of G_i in the image.
- C: center of curvature of the cornea.
- P: pupil center (coincident with the iris center).

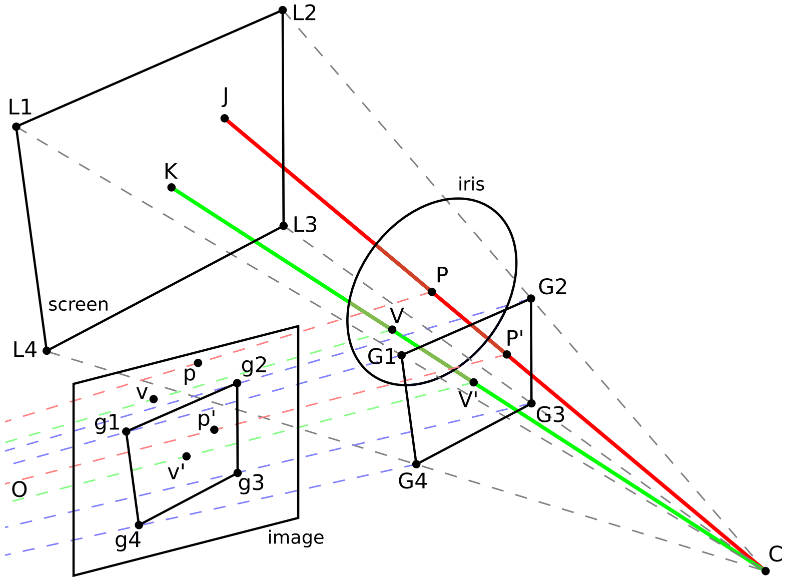


Fig. 5 Realistic geometric setup that should be considered for cross-ratio based eye gaze tracking

- J: intersection between the optical axis and Π_L.
- P': intersection between the optical axis with Π_G.
- V: intersection of the visual axis with the iris.
- K: intersection of the visual axis with Π_L (PoR).
- V': intersection of the visual axis with Π_G.
- p, p', v, v': images of P, P', V e V'.
- O: camera projection center.

Observing Fig. 5 it is possible to notice what happens if p and g_i are directly used to compute the PoR by application of the cross-ratio. First, p is the projection of P, a point that does not belong to Π_G. Consequently it is incorrect to assume that p and g_i are images of coplanar points. Besides that, the optical axis of the eye intercepts the screen at J, a point that does not correspond to the actual PoR (K).

To deal with these strong simplifying assumptions assumed by the **CR** method, new methods were developed based on the cross-ratio concept. These methods try, to a greater or lesser degree, to employ models that approximate to the complete scenario shown in Fig. 5. Some of these extensions will be introduced in the following sections, showing how gaze estimation error compensation is achieved, as well as pointing some issues still left open that motivated the work presented in this paper.

2.2.1 *Cross-Ratio with Multiple α Correction*

Yoo and Chung (2005) improved the **CR** method by correcting the error introduced in the gaze estimation due to the non-coplanarity of P and G_i. In their solution, the PoR is computed in the following way:

$$\text{PoR} = \mathbf{CR_f}\big(T_s(g_i, \alpha_i), p\big) \tag{9}$$

where T_s is a transformation defined by:

$$T_s(x, \alpha) = \alpha\,(x - g_0) + g_0 \tag{10}$$

In other words, T_s scales any image point x by α, relative to point g_0. This point is the image of the corneal reflection G_0, which is generated by a fifth light source that is attached near the camera's optical axis (note that when we refer to points g_i or G_i we are just considering the corneal reflections generated by the light sources attached to the screen corners). An important property of G_0 is that it belongs to the line $\overline{OC}$ and, as such, g_0 is the projection of C in the image plane.

The transformation of g_i in the image plane by T_s is equivalent to performing a scale of G_i in space (relative to C), so that G_i and P become coplanar, and then projecting these transformed points into the image plane.

Each point g_i has its own scale factor α_i. Because of this we will denote this method as the *Cross-ratio with Multiple α Correction* (**CR-M**α) method. These α_i values are obtained by a calibration procedure where a person has to look at each L_i point. Each α_i is computed as:

$$\alpha_i = \frac{\|p_i - g_0\|}{\|g_i - g_0\|} \tag{11}$$

where p_i corresponds to the image of the pupil center when the person gazes at L_i.

The idea behind this procedure lies in the fact that it is expected that p_i perfectly matches $T_s(g_i, \alpha_i)$ when the eye gazes at L_i. A problem with this approach is that, due to the κ angle between the optical and visual axes of the eye (not taken into account by the method), there is no guarantee that p_i will be in the line $\overline{g_i g_0}$. This way, the calibrated α_i parameters may not be accurate enough to compensate the non-coplanarity of P and G_i. Sum to that the lack of an explicit compensation of the κ angle, which leads to estimated gaze points being displaced from the actual observed points.

2.2.2 *Cross-Ratio with Displacement Vector Correction*

The *Cross-ratio with Displacement Vector Correction* (**CR-D**) method, developed by Coutinho and Morimoto (2006), is an extension of the **CR-M**α method in which the error introduced in the gaze estimation due to the κ angle is also compensated. For this method the PoR is computed by the following equation:

$$\text{PoR} = \mathbf{CR_f}\big(T_s(g_i, \alpha), p\big) + \mathbf{d} \tag{12}$$

or equivalently by:

$$\text{PoR} = \mathbf{CR_f}\big(g_i, T_s(p, \alpha)\big) + \mathbf{d} \tag{13}$$

Although in Coutinho and Morimoto (2006) the PoR is estimated using Eq. (12), we prefer to use the equivalent version of Eq. (13) because it is more suited to the geometric setup described in Fig. 5, in the sense that we want to bring P to Π_G in order to solve the non-coplanarity issue. In this version, instead of scaling all g_i points, just p is scaled. Since a single α value is used, it does not matter which set of points is scaled. Just keep in mind that the α used in Eq. (12) will be the inverse of the α used in Eq. (13).

When the suitable α value is used, the transformation of p by T_s results in an estimation of p', which is the projection of P', which in turn is the point where the optical axis intercepts Π_G. Since P' and Π_G are coplanar, the first source of error of the basic **CR** method is compensated.

The use of T_s is not enough, though, to accurately estimate the PoR. As can be seen in Fig. 5, the result of applying the $\mathbf{CR_f}$ function to image points g_i and p' is J, which is displaced from the actual PoR (K). To correctly compute K a displacement vector $\mathbf{d}$ must be added to J. The addition of $\mathbf{d}$ compensates the error in the gaze estimation introduced by the κ angle.

Parameters α and $\mathbf{d}$ are obtained by a calibration procedure where a person gazes at a set of on screen target points. Let X be the set of n calibration points and $Y^{\alpha c}$ the set of estimated PoRs for a given αc (α candidate) without the addition of any displacement vector. Let $\Delta^{\alpha c} = \{x_i - y_i^{\alpha c} \mid x_i \in X, y_i^{\alpha c} \in Y^{\alpha c}\}$ be the set of displacement vectors given by the difference between calibration and estimated points. Based on the observation that for the optimum α value vectors in Δ^{α} should be approximately constant, the optimum α will be the αc value that minimizes the following summation:

$$\sum_{i=1}^{n} \left\| \left(x_i - y_i^{\alpha c}\right) - mean\left(\Delta^{\alpha c}\right) \right\| \tag{14}$$

After the α parameter is computed, $\mathbf{d}$ is taken as the mean vector of the Δ^{α} set.

2.2.3 Homography Based Methods

Another approach to compensate the sources of error of the basic **CR** method is to use a *Homography* (**HOM**) transformation to map estimated gaze points (affected by both sources of errors) into the expected gaze points. This idea is presented by Kang et al. (2007) and Hansen et al. (2010). In both cases, the homographies used to correct the estimated gaze points are obtained by a calibration procedure during which a person has to gaze at some target points on the screen.

In Kang et al. (2007), the point of regard is computed by:

$$\text{PoR} = \mathbf{H_{LL}}\left(\mathbf{CR_f}(g_i, p)\right) \tag{15}$$

where $\mathbf{H_{LL}}$ is a homography that transforms estimated (incorrect) points in Π_L to expected (corrected) points in Π_L. Notice that no prior processing of the points passed as input to the $\mathbf{CR_f}$ function is performed.

An advantage of the homography mapping is that there is no need for the extra light source responsible for generating the corneal reflection G_0. The homography mapping can also be considered as a generalization of the transformations performed by the **CR-M**α (scale) and **CR-D** (scale and translation) methods, being able to also correct perspective distortions.

In the homography method presented in Hansen et al. (2010) the PoR is computed by:

$$\text{PoR} = \mathbf{H_{NL}}\left(\mathbf{CR_n}(g_i, p)\right) \tag{16}$$

The function $\mathbf{CR_n}$ is a variation of the $\mathbf{CR_f}$ function in which the returned point is computed relative to a unitary square (normalized space), instead of being relative to the rectangle formed by L_i points. The homography $\mathbf{H_{NL}}$ then transforms estimated gaze points in the normalized space to expected gaze points in the screen space (Π_L).

The use of a normalized space adds another advantage to the homography method: the dimension of the rectangle formed by L_i does not need to be known. When the normalized space is not used and dimensions of L_i needs to be known, conversions between metric unit (physical size of the rectangle) and pixel unit must take place, during which eventual offsets between the L_i rectangle and the useful screen area must also be taken into account. This way, the use of the normalized space facilitates implementation, by dissociation of the Π_L plane from the plane over which we want to track a person's gaze.

3 Head Movement Tolerance of CR Based Methods

The extensions to the **CR** method previously presented (Yoo and Chung 2005; Coutinho and Morimoto 2006; Kang et al. 2007; Hansen et al. 2010) were successful, to a greater or lesser degree, in compensating the error introduced in gaze estimations due to the strong simplifying assumptions assumed by the **CR** method in its basic form. In particular, the compensation performed by the **CR-M**α method is incomplete since it compensates the non-coplanarity of P and G_i, but neglects the compensation of the κ angle. The **CR-D** and **HOM** methods, on the other hand, fully compensates both sources of error.

Error correction strategies employed by these extensions include: the transformation of the input arguments to the $\mathbf{CR_f}$ function (the case of the **CR-M**α method); the transformation of the value returned by the $\mathbf{CR_f}$ function (the case of the **HOM** methods); and the transformation of both input arguments and value returned by the $\mathbf{CR_f}$ function (the case of the **CR-D** method). Despite the different strategies and different implementations, all of these transformations rely on calibrated parameters that minimize the gaze estimation error for a set of screen targets used as calibration points.

Among the two sources of error of the basic **CR** method (non coplanarity of P and G_i, and the κ angle between the visual and optical axes of the eye), the κ angle is the one that most contributes to the lack of robustness to head movement that is observed in the extensions of the **CR** method. This is illustrated on Fig. 6 that shows the values of the α and $\mathbf{d}$ parameters used by the **CR-D** method that were obtained after calibrating the method at different head positions (the data shown in the figure were obtained by simulation whose setup is detailed in Sect. 6.1). From Fig. 6 it is clear that the value for α is quite stable for all head positions, while the component values of $\mathbf{d}$ show a great variation across different positions.

Although the **CR-D** method was used to illustrate how the optimal value of $\mathbf{d}$ changes according to the head position, a similar effect occurs with the **HOM** methods. For this methods some of the coefficients of the homography transformation used to correct the gaze estimation will also show a similar behavior to what was observed for $\mathbf{d}$. Based on this

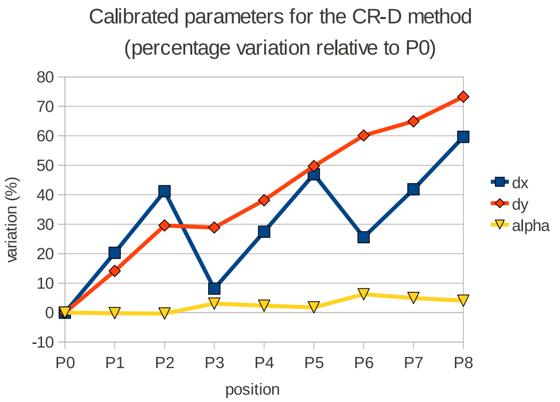


Fig. 6 Calibrated parameters (α and $\mathbf{d}$) for the **CR-D** method at different head positions (P_0 through P_8). Shown in the graph are the percentage variation of the calibrated parameters relative to P_0. Note that the **CR-D** method was calibrated at positions P_0 through P_8 for the purpose of illustrating the variation of the calibrated parameters. In practice, we want to perform calibration at just one position

observation we can note that the calibrated parameters used by the **CR** extensions to explicitly correct the κ angle are not suitable to correct the gaze estimation at arbitrary head positions. In fact, the calibrated parameters are optimized for the specific head position where calibration is performed, which somewhat imposes limits on the freedom of head movement.

This work was motivated by the observation that head movement tolerance could still be improved for gaze estimation based on the cross-ratio principle. As a result, we present two new methods: the *Cross-ratio with Dynamic Displacement Vector Correction* (**CR-DD**) method and the *Planarization of CR Features* (**PL-CR**) method. The main idea of the **CR-DD** method is to estimate changes in head position in order to adjust calibrated parameters accordingly. The **PL-CR** method follows a different approach, employing a set of calibrated parameters that are invariant to head movement. Each of these methods are presented in greater detail in Sects. 4 and 5. Before introducing them, for better understanding of how κ affects gaze estimation, under the condition of head movement, a more detailed investigation is presented in the following section.

3.1 The Influence of κ on Gaze Estimation

To illustrate how κ affects gaze estimation, consider two simple scenarios shown in Fig. 7. C_t correspond to the position of the cornea center, J_t the intersection of the optical axis with the screen, and K_t the intersection of the visual axis with the screen. Assume that the values C_0, J_0, and K_0 are computed at a fixed calibration position. The first scenario, on the left of Fig. 7, shows a depth translation of the eye, and the second scenario, on the right of Fig. 7, shows a rotation. For simplicity, consider that the optical axis is perpendicular to the screen at the calibration position in both scenarios.

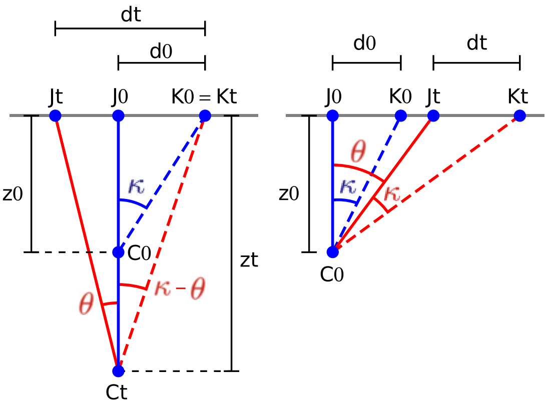


Fig. 7 Variation of the on-screen offset d_t due to eye translations and rotations

At the calibration position, κ can be computed as:

$$\kappa = \operatorname{atan}\left(\frac{d_0}{z_0}\right) \tag{17}$$

where d_0 is the displacement between J_0 and K_0 (respective intersections of the optical and visual axes with the screen when the cornea is at C_0), and z_0 is the distance of C_0 to the screen.

As the eye gets farther from the screen, to keep the gaze on the same screen position, the eye has to rotate by an angle θ. Assuming that κ is a constant eye parameter, the offset between the intersections of the optical and visual axis is now d_t. Since most gaze methods are only able to compute J_t, if a constant offset, such as d_0, is used to compensate for κ (as suggested in Coutinho and Morimoto 2006), then the new gaze position would be computed as:

$$K'_t = J_t + d_0 \tag{18}$$

which is different than the true K_t position shown in Fig. 7. Therefore the error contribution due to this constant offset would be:

$$\begin{aligned} \epsilon &= \left\| K_t - K'_t \right\| = \| d_t - d_0 \| \\ &= \| J_t - J_0 \| = z_t \tan(\theta) \end{aligned} \tag{19}$$

As z_t goes to infinity, the visual axis becomes perpendicular to the screen, and θ becomes equivalent to κ. This results shows that the methods that use a constant offset have an upper bound on the estimated gaze error due to translation equal to κ.

The second scenario shows a rotation around C_0. Assuming once again that κ is a constant eye parameter, a rotation of the optical axis by θ would move J_0 to J_t. From the geometry shown in Fig. 7, K_t and K'_t can be computed as:

$$\begin{aligned} K_t &= z_0 \tan(\theta + \kappa) + J_0 \\ K'_t &= J_t + d_0 = z_0 \tan(\theta) + J_0 + d_0 \end{aligned} \tag{20}$$

and therefore, assuming J_0 as a reference point, we can compute the estimation error due to a rotation by θ as:

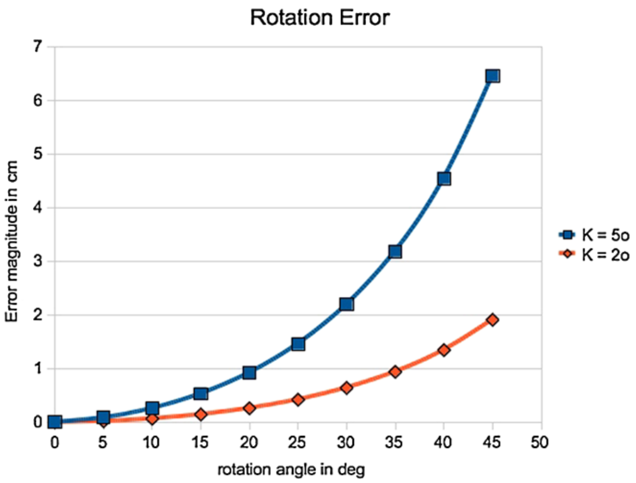


Fig. 8 Variation of the on-screen error of a constant offset method due to eye rotation, assuming the eye is 60 cm far from the screen

$$\begin{aligned}\epsilon_\theta &= K_t - K_t' \\ &= z_0\big(\tan(\theta+\kappa) - \tan(\theta) - \tan(\kappa)\big) \\ &= z_0\frac{\tan(\theta)\tan(\kappa)(\tan(\theta)+\tan(\kappa))}{1-\tan(\theta)\tan(\kappa)} \qquad (21)\end{aligned}$$

Observe that for large values of θ, the offset contribution due to κ becomes larger and it is not bounded since the visual axis can be parallel to the screen. Figure 8 illustrates how the rotation error behaves for $z_0 = 60$ cm, assuming $\kappa = 5°$ and $\kappa = 2°$. Assuming a 19″ monitor and the eye position directly in front of the center of the screen, the eye would need to rotate about 18° to cover the whole screen. If the eye is positioned towards the edge of the screen, it would have to rotate about 36° to look at the other end. Observe from Fig. 8 that the error for $\kappa = 5°$ for a 20° rotation is about 1 cm (approximately 1° of the visual angle) and about 3 cm for a 35° rotation. For $\kappa = 2°$, the influence on the error magnitude is much smaller.

These results show that translations of the eye parallel to the screen, that would require a rotation of the eye relative to the calibration position, may cause large estimation errors towards the edges of the screen, when a constant offset method is applied.

Also observe that for eyes with small κ the influence of the correction mechanisms on the gaze estimate will be smaller.

4 CR-DD: Cross-Ratio with Dynamic Displacement Vector Correction

The **CR-D** method (Coutinho and Morimoto 2006), previously introduced, treats both sources of error of the **CR** method pointed by Guestrin et al. (2008). However, as verified in Coutinho and Morimoto (2006), some accuracy decay in gaze estimation is observed under head movements, mainly depth movements, i.e., movements of the head in the direction perpendicular to the screen.

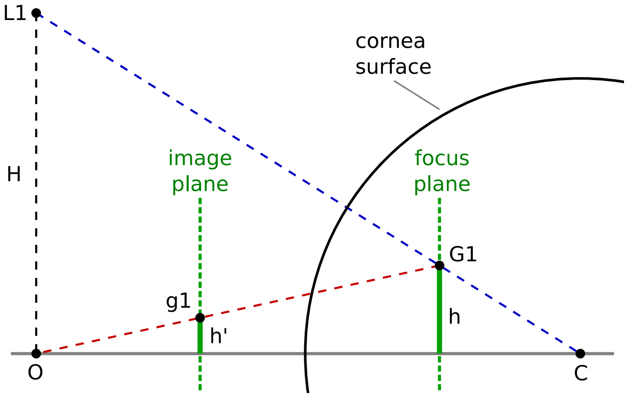


Fig. 9 Image formation of a corneal reflection generated by a light source

The goal of the **CR-DD** method is to extend the **CR-D** method to improve gaze tracking accuracy under the particular case of depth movements, the type of head movement that most affects the **CR-D** method. The previous analysis of the influence of κ on gaze estimation results showed why depth movements increase gaze estimation error.

If it is possible to measure the eye distance to the screen, it is possible to adjust $\mathbf{d}$ so that its length is adequate to the eye distance in a given moment, thus minimizing error. This solution is not ideal, since the length and orientation of $\mathbf{d}$ are functions of both eye distance and rotation, but it is possible to compensate the portion of the error introduced due to eye translations in the direction perpendicular to the screen.

Consider $\mathbf{d_0}$ the reference displacement vector obtained by the calibration procedure of the **CR-D** method, which was executed at a reference distance z_0. As $\|\mathbf{d_t}\|$ is directly proportional to current distance z_t (as shown in Eq. (17)), a more suitable displacement vector $\mathbf{d_t}$ for an arbitrary distance z_t can be computed by:

$$\mathbf{d_t} = \left[\frac{z_t}{z_0}\right]\mathbf{d_0} \qquad (22)$$

4.1 Estimating Distance Variation

In order to compute the displacement vector as indicated by Eq. (22), both the reference distance z_0 and the current distance z_t need to be known. Alternatively, by observation of the size of the quadrilateral formed by g_i (corneal reflections in the image), it is possible to estimate the ratio z_t/z_0 without needing to know the absolute values of z_t and z_0.

Figure 9 illustrates the image formation of a corneal reflection for a single light source. Let L_1 be the light source, O the camera center of projection, C the center of curvature of the cornea and z the distance from O to C. Consider also that the camera has focal length f_{cam}, that the cornea surface is a convex spherical mirror of focus f_{cor}, G_1 is formed at the focus plane and projected onto the image plane at g_1. Heights h and h' (from G_1 and g_1, respectively, relative to the camera axis) are given by:

$$h = \frac{f_{cor} H}{z}, \tag{23}$$

$$h' = \frac{f_{cam} h}{z - f_{cor}} \tag{24}$$

Substituting (23) in (24), we have that:

$$h' = \frac{f_{cam} f_{cor} H}{(z - f_{cor}) z} \rightarrow h' \sim \frac{f_{cam} f_{cor} H}{z^2} \tag{25}$$

Using the approximation for h' (justified by the fact that distance f_{cor} is much smaller than z), and considering two different distances z_t and z_0 with respective heights h'_t and h'_0, it is possible to compute the ratio z_t/z_0 in the following way:

$$\frac{z_t}{z_0} = \sqrt{\frac{h'_0}{h'_t}} \tag{26}$$

Therefore, by measuring the variation in the size of the quadrilateral formed by g_i, it is possible to estimate the relative translation of the eye from an initial reference position (z_0), and the displacement vector $\mathbf{d_0}$ can be adjusted accordingly.

The **CR-DD** method uses the same calibration procedure of the **CR-D** method with a few additions. At calibration distance z_0, besides computation of the α scale factor and the displacement vector $\mathbf{d_0}$, we also compute the reference size $size_0$ of the quadrilateral formed by g_i. The size of quadrilateral was taken as the sum of its diagonal lengths. Since during calibration a person looks to different points across the screen, $size_0$ is computed as the average value of all quadrilateral sizes measured for all calibration points.

After calibration of α, $\mathbf{d_0}$ and $size_0$, gaze estimation at distance z_t is performed in the following way:

$$\text{PoR} = \mathbf{CR_f}\big(g_i, T_s(p, \alpha)\big) + \frac{z_t}{z_o}\mathbf{d_0} \tag{27}$$

which is equivalent to:

$$\text{PoR} = \mathbf{CR_f}\big(g_i, T_s(p, \alpha)\big) + \sqrt{\frac{size_0}{size_t}}\mathbf{d_0} \tag{28}$$

5 PL-CR: Planarization of CR Features

Recall Fig. 5 that illustrates a more realistic geometric setup for the cross-ratio based methods for remote eye gaze tracking. For this scenario, the basic cross-ratio principle presented in Sect. 2.2 cannot be directly applied to estimate the PoR(K). Observing Fig. 5 we can see that p is projection of P, a point that does not belong to Π_G and as such it is incorrect to assume that p and g_i are images of coplanar points. Also, the intersection of the optical axis with the screen plane (J) does not corresponds to the point that the eye is actually gazing (K).

It is straightforward to see that if point v' can be computed, and v' and g_i are used to compute K using the basic cross-ratio principle then all sources of error regarding the geometric setup will be eliminated. Remember that v' is the image of V' the point where the visual axis intercepts the plane Π_G. Therefore the use of V' satisfies the two simplifying assumptions assumed by the basic cross-ratio method: V' and G_i are coplanar and V' is a point in the line of sight. For the **PL-CR** method the PoR is computed in the following way:

$$\text{PoR} = \mathbf{CR_f}\big(g_i, v'\big) \tag{29}$$

The challenge of the **PL-CR** method is to find a way to estimate v' given image points p and g_i. Since v' is the projection of V', which in turn is defined by the intersection of $\overrightarrow{CV}$ with Π_G, our problem resumes to the estimation of $\overrightarrow{CV}$ and Π_G. Because points G_i are not exactly coplanar, when we compute Π_G, what is actually computed is an approximate plane such that the distances from G_i to the plane are minimized.

To keep the hardware requirements of the **PL-CR** method simple (same configuration of other **CR** based methods), we will assume a weak perspective camera model for estimation of $\overrightarrow{CV}$ and Π_G. In the weak perspective camera model, image formation can be described by an orthographic projection, followed by a scale (Emanuele Trucco 1998). The use of such camera model is justified by the fact that the size of the eye (our object of interest) is much smaller than the typical distance from the eye to the camera. For the solutions to these sub problems, the scale component of the weak perspective model is not relevant since estimation of $\overrightarrow{CV}$ and Π_G do not take place in real world metrics. This way, a simpler orthographic camera model can be assumed.

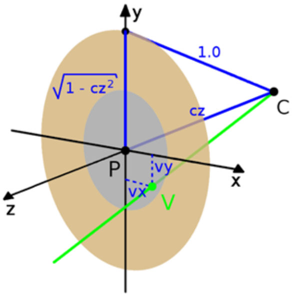
Before following to the presentation of the solutions to each sub problem, we will first introduce the eye model and relevant coordinate systems considered by the **PL-CR** method.

5.1 Eye Model

In order to reconstruct the visual axis in 3D space and compute its intersection with Π_G, we will consider the eye model that is shown in Fig. 10. For this model, consider the following orthonormal coordinate system: origin at pupil/iris center P, plane xy coincident with the iris plane, with y axis pointing in the upward direction, x in the horizontal direction and z perpendicular to the iris (corresponding to the optical axis of the eye).

Relevant points for this model are the cornea center C and the point V where the visual axis intercepts the pupil/iris

Fig. 10 Normalized eye model



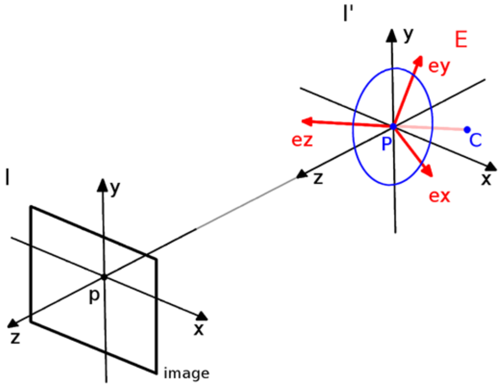


Fig. 11 Relevant coordinate systems for the **PL-CR** method

plane. C belongs to the z axis and its coordinates are given by $(0, 0, -c_z)$. V belongs to the xy plane and has coordinates $(v_x, v_y, 0)$. This model has, therefore, 3 parameters (v_x, v_y and c_z) that will be estimated by a calibration procedure. Similar to other gaze tracking methods, the calibration procedure consists of finding values for v_x, v_y and c_z that minimize the gaze estimation error for a set of calibration points. Since the model parameters are independent on eye location, the calibration procedure needs to be done just once per person and also ensures robustness to head movements.

This model is a normalized model where the cornea radius has a value of 1.0. This way, the iris radius is given by $\sqrt{1 - c_z^2}$. The use of a normalized eye eliminates the need to know the absolute values of the eye structures. What is important, in this case, are the ratios between model elements.

5.2 Coordinate Systems

Besides the normalized eye model, it is also important to define 3 orthonormal coordinate systems, shown in Fig. 11: the image coordinate system **I** (represented by the $\mathbf{F_I}$ matrix), the translated image coordinate system **I**′ (represented by the $\mathbf{F'_I}$ matrix) and the eye coordinate system **E** (represented by the $\mathbf{F_E}$ matrix).

Coordinate system **I** has its xy plane coincident with the image plane, z axis perpendicular to xy, origin in p, and units given in pixels. Coordinate system **I**′ has x, y and z axes equal to those from **I**, with origin in P.

Since an orthographic camera model is being used, the projection of a given point in the image plane is equivalent to the projection of the corresponding point in the plane xy of **I**′. The distance between the origins of **I** and **I**′ are unknown and can have any arbitrary value, being not relevant to the **PL-CR** method. We will assume **I**′ to be our reference coordinate system. Estimation of the visual axis and the plane Π_G will take place relative to this reference system. This way, any point that does not have an explicit indication of a coordinate system are assumed to be relative to **I**′.

Coordinate system **E** is also centered at P with its orthonormal axes defined by:

$$\mathbf{e_z} = \frac{\mathbf{n}}{\|\mathbf{n}\|}, \tag{30}$$

$$\mathbf{e_x} = \frac{\mathbf{up} \times \mathbf{e_z}}{\|\mathbf{up} \times \mathbf{e_z}\|}, \tag{31}$$

$$\mathbf{e_y} = \mathbf{ez} \times \mathbf{e_x} \tag{32}$$

where **n** is the normal to the iris (it represents the optical axis of the eye) and the **up** vector is a reference to the world vertical direction. Without this reference, there would be infinite possibilities for the $\mathbf{e_x}$ and $\mathbf{e_y}$ vectors of coordinate system **E**, and consequently infinite possibilities for the V_E point when transformed to the reference coordinate system **I**′. Estimation of the **up** vector and **n** will be detailed in the following sections.

5.3 Visual Axis Estimation

Contrary to what happens for the optical axis, for which there is the pupil center, there is no visible eye structure associated with the visual axis. One could argue that the pupil center is not a visible structure as well, but from the pupil border it is possible to have a good estimate of its center.

It is due to this lack of something “visible” that the eye model presented is important. Having knowledge of the normalized model, we can compute C and V from the observed iris pose from an eye image. Estimation of the visual axis consists of finding the coordinates of C and V in the reference coordinate system **I**′. C and V can be computed by the following formulas:

$$C = s\mathbf{F_E}C_E, \tag{33}$$

$$V = s\mathbf{F_E}V_E \tag{34}$$

where $C_E = (0, 0, -c_z)$, $V_E = (v_x, v_y, 0)$ and s is a scale factor given by:

$$s = \frac{r_t}{\sqrt{1 - c_z^2}} \tag{35}$$

that has the role of scaling the normalized eye model so that its dimensions match the dimensions of the eye in the image at a given time instant t, with r_t being the iris radius (in pixels) at t. Despite the fact that the iris can have an elliptical shape in the image, its radius is given by the length of its major semi-axis.

The **up** vector used to define the **E** coordinate system is a reference to the real world vertical direction. The $(0,1,0)$ vector in the $\mathbf{I}'$ coordinate system may not correspond to the real world vertical direction if the camera is pointed upwards, downwards or is rotated around its optical axis. Assuming that the screen plane is parallel to the world vertical direction, the **up** vector can be inferred from the positions of light sources L_i by:

$$\mathbf{up} = \frac{(L_1 - L_4) + (L_2 - L_3)}{\|(L_1 - L_4) + (L_2 - L_3)\|} \tag{36}$$

with L_i values expressed in the $\mathbf{I}'$ coordinate system. As will be shown in Sect. 5.5, L_i are computed before Π_G estimation and can be used to compute the **up** vector. However, as will also be shown in Sect. 5.5, in order to estimate L_i, C must be known, and computation of C depends on the **up** vector to define the $\mathbf{F_E}$ matrix. How to compute C, then? Since C lies in the $\mathbf{e_z}$ direction, any arbitrary **up** vector can be used to compute it. For estimation of C we assume $\mathbf{up} = (0,1,0)$ and after L_i are estimated **up** is recalculated using Eq. (36). After the correct value of **up** is computed, V can finally be estimated.

5.4 Iris Normal Estimation

Estimation of the iris normal $\mathbf{n}$ is a prerequisite to define the transformation matrix $\mathbf{F_E}$, which in turn is used to estimate the visual axis. A solution that estimates the iris normal based on the observed iris contour is presented in the work by Wang and Sung (2002). Because iris edge detection is more difficult than pupil detection, mostly due to occlusion by the upper and lower eyelids, we used a different approach for iris normal estimation. Instead of relying on the iris edge, this approach is based of the pupil center P, the c_z parameter of the eye model, and the corneal reflection G_0 (which is generated by the light source placed at the optical axis of the camera).

Consider the points G_0 and C and their projections on the image g_0 and c. Assuming the cornea as a spherical surface, it is known that O, G_0 and C are collinear and consequently c coincides with g_0. Since the iris normal is given directly by $\mathbf{n} = P - C$, the projection of $\mathbf{n}$ in the image will be $\mathbf{m} = p - g_0$. Remembering that an orthographic camera model is assumed, the n_x and n_y components of $\mathbf{n}$ can be extracted from the image as $n_x = m_x$ and $n_y = m_y$. Therefore, n_z is the only missing component of $\mathbf{n}$, whose module is expressed by:

$$|\mathbf{n}| = \sqrt{n_x^2 + n_y^2 + n_z^2} \tag{37}$$

Using the scale factor s previously introduced, it is also known that:

$$|\mathbf{n}| = s c_z \tag{38}$$

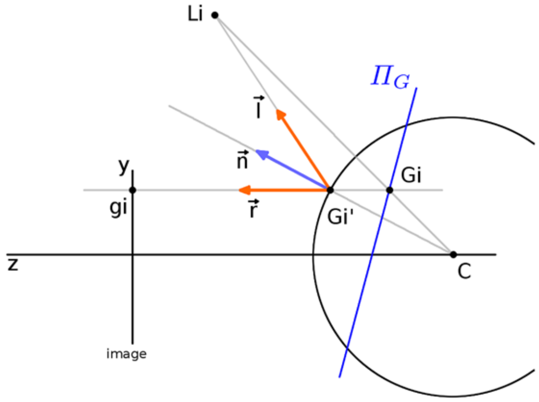


Fig. 12 Corneal reflection formation, assuming orthographic projection. Also depicted in the figure are the point G_i and plane Π_G

Combining (37) and (38), n_z can be computed by:

$$n_z = \sqrt{s^2 c_z^2 - n_x^2 - n_y^2} \tag{39}$$

to finally obtain the iris normal $\mathbf{n}$.

5.5 Plane Π_G Estimation

To estimate Π_G, we need to find the plane in space such that the projections of L_i on this plane, with C being the projection center, are equivalent to G_i. This implies that G_i belongs to the line segment $\overline{L_i C}$. Also, if G_i' is the point on the spherical cornea surface where the specular reflection due to L_i occurs and G_i' is projected to the image point g_i, then g_i, G_i' and G_i are collinear points. This way, two lines containing G_i are defined, and G_i can be computed by their intersection (see Fig. 12). It is worth noting that the intersection of two lines in 3D space can have a complicating factor because they hardly intersect at an exact point. To avoid this problem, the middle point of the smallest line segment connecting the two lines is computed as the intersection.

Computation of three among the four G_i points is sufficient to estimate the plane Π_G (the plane over which all light sources L_i are projected, with C being the projection center). However, since the surface that contains all G_i points is not exactly planar, Π_G is computed as the approximate plane such that the summation of the distances between G_i and Π_G are minimized.

To compute each G_i point, two lines passing through it must be defined. The line defined by g_i and G_i' is simple to be described considering the orthographic camera assumption. By this hypothesis, coordinates x and y of g_i, G_i' and G_i are the same, and the vector representing the reflected light ray is given by $\mathbf{r} = (0,0,1)$. The first line is then defined by:

$$R_i : g_i - a_i \mathbf{r} \tag{40}$$

with x and y coordinates of g_i being extracted directly from the image. For the z coordinate any arbitrary positive value bigger than the cornea radius can be defined to ensure that g_i is in front of the eye. In the line equation, a_i is a coefficient that represents the distance between a point in the line and g_i.

The second line is defined by C and L_i but, for now, just C is known. In order to define this second line, L_i must be computed as well. L_i can be expressed by the following equation:

$$L_i = G'_i + b_i \mathbf{l_i} \tag{41}$$

where: $\mathbf{l_i}$ is the vector that goes in the opposite direction of the light ray reaching G'_i (see Fig. 12); G'_i can be computed by the intersection of R_i with the cornea surface (a sphere of radius s, centered in C); and finally, $\mathbf{l_i}$ can be obtained by reflecting $\mathbf{r}$ at G'_i.

If each equation for L_i is taken individually, it is not possible to compute L_i because b_i remains unknown at each equation. However, from knowledge of the distances between the L_i points (i.e., knowledge of the dimensions of the rectangle formed by L_i), an overdetermined system of six equations with four unknowns (b_1, b_2, b_3, and b_4) can be defined and solved, for example by least squares minimization. This system of equations is described in (42).

$$\begin{aligned}
&\langle L_1 - L_2, L_1 - L_2\rangle = w^2,\\
&\langle L_2 - L_3, L_2 - L_3\rangle = h^2,\\
&\langle L_3 - L_1, L_3 - L_1\rangle = d^2,\\
&\langle L_3 - L_4, L_3 - L_4\rangle = w^2,\\
&\langle L_4 - L_1, L_4 - L_1\rangle = h^2,\\
&\langle L_2 - L_4, L_2 - L_4\rangle = d^2
\end{aligned} \tag{42}$$

The values of w, h and d in the system correspond to, respectively, the width, height and diagonal of the rectangle formed by L_i. Note that in our reference coordinate system $\mathbf{I}'$ units are given in pixels and therefore the values of w, h and d must be expressed in pixels as well. Conversion of such values from metric space to pixel space can be accomplished by:

$$value_p = s\frac{value_m}{r_m} \tag{43}$$

where $value_p$ is measured in pixels, $value_m$ in an arbitrary metric unit, r_m is the cornea radius in the same arbitrary metric unit, and s the scale factor that is equivalent to the cornea radius in pixels. For the value of r_m we used the average value of 0.78 cm.

Once the equation system is solved and L_i are computed, we are able to define the line $\overline{L_i C}$ and compute its intersection with R_i, thus obtaining G_i. With G_i, the approximate plane Π_G can finally be estimated.

5.6 v' Estimation

Once Π_G, V and C are computed, estimation of v' is straightforward. First V' is computed as the intersection of $\overline{CV}$ with Π_G. Next we project V' to the image plane. Since an orthographic camera model is used $v' = (V'_x, V'_y, 0)$.

6 Evaluation of the Proposed Methods

To evaluate the performance of the proposed **CR-DD** and **PL-CR** methods, and compare them to other cross-ratio based methods (**CR-D** and **HOM**), simulations and user experiments were conducted. To facilitate analysis and discussion of the results, in the remainder of this paper we will define two groups of methods being tested. The first group is defined by the methods that apply some kind of head movement compensation (HMC methods), and includes the **CR-DD** and **PL-CR** methods. The second group includes the methods that do not explicitly perform head movement compensation (non-HMC methods) and includes the **CR-D** and **HOM** methods (the **HOM** method tested is the one described in Hansen et al. (2010) without the Gaussian process error modeling).

For both simulations and users experiments, evaluation consisted of measuring the gaze estimation error (in degrees) at different head positions. At each head position, the subjects (or the simulated eye) had to gaze at a group of screen targets and the gaze estimation error for each observed target was computed using the following formulas:

$$\mathbf{t} = \frac{T - S}{\|T - S\|}, \tag{44}$$

$$\mathbf{k} = \frac{K - S}{\|K - S\|}, \tag{45}$$

$$\epsilon = \frac{180 \cos^{-1}(\langle \mathbf{t}, \mathbf{k}\rangle)}{\pi} \tag{46}$$

where T is the observed target, K the estimated gaze point and S the subject's head position. The average gaze estimation error for a particular head position was computed by:

$$\frac{1}{N}\sum_{j=1}^{N} \epsilon_j \tag{47}$$

where N is the number of screen targets used as test points and ϵ_j the gaze estimation error observed when the eye was gazing at the target point number j.

6.1 Simulation Setup

For the simulations, synthetic images generated by ray tracing were used. The LeGrand eye model was adopted for image generation and its measures were extracted from the table compiled in Guestrin and Eizenman (2006). In this model the cornea and the aqueous humour are combined into a single medium (with index of refraction of 1.3375) so that refraction occurs only at the external cornea surface. The cornea has a radius of 0.78 cm and the pupil center is located 0.42 cm from the cornea center. During simulations, when the eye was directed to a given target, it was the visual axis of the eye that effectively intercepted the target. Two

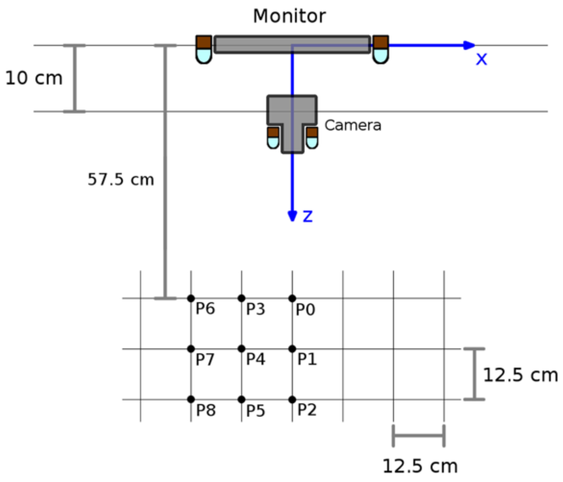


Fig. 13 Layout of head positions used in simulations and user experiments. P_0 was used as the calibration position for all gaze estimation methods being compared

configurations for the visual axis were used. The first had horizontal and vertical angle values of 5° and 1.5° respectively. In the second configuration a horizontal angle of 2° and a vertical angle of 0.6° were used.

The simulated screen consisted of a rectangle of 34 by 27 cm, with light sources L_i positioned at each screen corner. The camera was positioned below the bottom border of the screen. More precisely, it was located 2 cm below the middle point of the bottom border. The central point of the screen is the origin of the coordinate system considered for the simulations, with x and y axes corresponding to the horizontal and vertical directions, and z axis perpendicular to the screen plane. A set of 49 test targets, arranged in a regular 7×7 grid, was used. For calibration of each gaze estimation method, a subset of 9 points from the 49 test targets, arranged in a regular 3×3 grid, was used.

A perspective camera model with a vertical field of view of 5° was considered to generate the images used by the simulations. Because of its limited field of view, for each head position the simulated camera had to be rotated to point in the direction of the cornea center. This camera configuration was chosen to replicate the characteristics of the actual camera used in the user experiments.

The layout of head positions used for simulations, as well as for the user experiments, is shown in Fig. 13. A total of 9 head positions were used, with the following coordinates: $P_0 = (0, 0, 57.5)$, $P_1 = (0, 0, 70)$, $P_2 = (0, 0, 82.5)$, $P_3 = (-12.5, 0, 57.5)$, $P_4 = (-12.5, 0, 70)$, $P_5 = (-12.5, 0, 82.5)$, $P_6 = (-25, 0, 57.5)$, $P_7 = (-25, 0, 70)$, $P_8 = (-25, 0, 82.5)$. This set of positions represents lateral, depth and combination of lateral and depth head movements. It is important to note that the camera position depicted in Fig. 13 reflects the camera positioning used in the user experiments. For the simulations, as already stated, the camera lies on the y axis, below the bottom screen border.

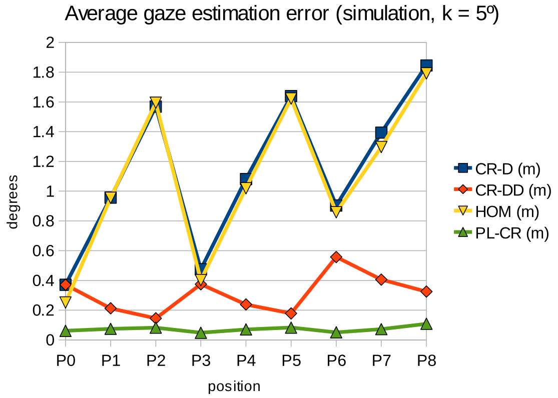


Fig. 14 Average gaze estimation error for simulation of cross-ratio based methods, with $\kappa = 5°$

Since the main objective of the evaluation is to verify how each extension to the **CR** method behaves under the condition of head movement, only position P_0 was used to calibrate all gaze estimation methods being evaluated (this applies not only to the simulations but also to the user experiments). All gaze estimation results for other head positions (P_1 through P_8) are computed using the set of calibrated parameters obtained at P_0.

6.2 Simulation Results

Simulation results are presented in Figs. 14 and 15. Figure 14 shows the results for the first configuration of visual axis (horizontal rotation angle of 5°) while in Fig. 15 results for the second configuration (horizontal angle of 2°) are shown. Each graph presents the average gaze estimation error for all methods being compared (**CR-D**, **HOM**, **CR-DD** and **PL-CR**) at each head position (P_0 through P_8). The graph's vertical axis corresponds to the visual angle error in degrees. The horizontal axis corresponds to each head position.

As expected, HMC methods present a better performance (smaller average gaze estimation error) than non-HMC methods as the eye moves away from the calibration position (P_0) for both simulation conditions (κ values of 5° and 2°). The major observed difference between the two conditions is that accuracy decay for the non-HMC methods is directly proportional to the magnitude of κ. For $\kappa = 5°$, the maximum error observed for all methods and positions is about 1.85° of visual angle, while for $\kappa = 2°$, a maximum error of about 0.77° is observed. This observation indicates that for subjects with smaller κ angles, the improvements of the HMC methods will be less noticeable than for subjects with larger κ values.

If we consider just the non-HMC methods, it is possible to note that they are more affected by translations in z

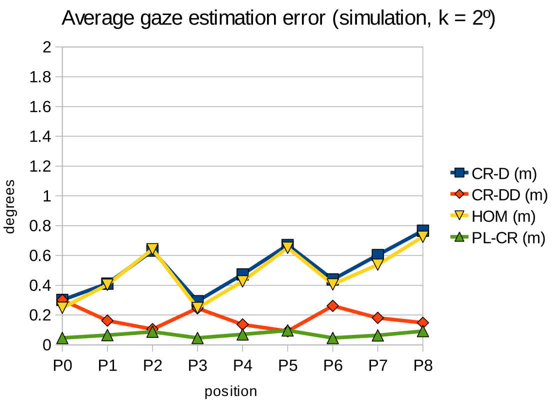


Fig. 15 Average gaze estimation error for simulation of cross-ratio based methods, with $\kappa = 2°$

(depth head movements) than translations in x (lateral head movements). This is related to how κ affects gaze estimation results as discussed in Sect. 3.1, where it was shown that the distance between the eye and the screen plays a major role on the error due to κ. A comparison between the **CR-D** and **HOM** methods shows a better performance of the **HOM** method, specially at the calibration position. This is explained by the fact that the homography correction is more flexible than the scale and translation compensation used by the **CR-D** method.

When we compare the results for the HMC methods a difference in performance between them can also be observed. The **PL-CR** method achieves better gaze estimation accuracy (maximum error of 0.11° considering both values of κ) than the **CR-DD** method (maximum error of 0.56° for $\kappa = 5°$ and 0.30° for $\kappa = 2°$). Results for the **PL-CR** method are also more stable across all head positions when compared to the **CR-DD** method.

This difference between the two HMC methods is explained by the different approaches taken by each. Although the **CR-DD** method compensates head movement by measuring the variation of the distance between the eye and the screen, the compensation applied is incomplete, as eye rotation is not taken into account. Starting at calibration position P_0, when the eye translates in the x direction, the overall distance from the eye to the screen does not change significantly, while it will need to rotate more to be able to gaze all regions of the screen. On the other hand, if the eye translates in z by the same amount (going farther from the screen), the distance between the eye and the screen will increase by the translation amount, at the same time that the eye will need to rotate less to be able to gaze all screen points. This characteristic makes the **CR-DD** method effective in compensating translations in z, but not as effective in compensating translations in x, a behavior that is confirmed by the simulation results. It possible to note from the results that the performance difference between the **CR-DD** and **PL-CR** methods is more evident for head positions closer to the screen (P_0, P_3, and P_6), while for positions farther to the screen (P_2, P_5, and P_8), the performance of both methods are more similar.

The **PL-CR** method for estimation of v' compensates both aspects of eye movement (position and rotation), which explains the better performance and the smaller variation in the gaze estimation error at all head positions. This performance difference between the **CR-DD** and **PL-CR** methods shows the importance of considering eye rotations to proper compensate head movements. Nevertheless, the **CR-DD** method is a simple solution that is capable of improving gaze estimation under a particular condition of head movements. The **PL-CR** method, on the other side, requires extra computation to estimate v', which might be a potential source of error.

6.3 Experimental Design

To be able to compare all methods using the same user data, the data was first collected and then processed offline. The collected data consists of images of the eye that were captured while the subjects gazed at the test points. In general, experimental conditions were very similar to those described for the simulations, including the use of the layout of head positions shown in Fig. 13. There are some important differences, however, that should be pointed out.

The first is related to the screen (by screen we mean the visible area of the monitor). Both the simulated and real screen used in the experiment have the same dimensions (34 by 27 cm), but light source positioning varied between the two conditions. In the simulations, light sources exactly matched the screen corners. For the experiment it was not possible to exactly fixate the light sources (infrared LEDs) on the screen corners due to the monitor's border. Nevertheless, care was taken to ensure that the infrared LEDs were aligned with the screen plane. Also note that the monitor's borders were added to the screen dimensions to determine the correct size of the rectangle defined by L_i.

The camera position relative to the screen also varied between the simulation and experimental conditions. In the simulations the camera was placed exactly below the bottom border of the screen. For the user experiment the camera was placed 10 cm towards the user as can be seen in Fig. 13. The camera used for the experiments had a manual focus and a narrow field of view (about 5°). Therefore, for each different head position the camera had to be directed to the subject's eye and its focus adjusted.

A camera with a narrow field of view was chosen for the user experiment (as well as for the simulations) because it permits the capture of detailed close-up images of the subject's (and simulated) eyes for an image resolution of 640 × 480 pixels. This camera setup ensures that the size (in pixels) of the quadrilateral formed by g_i is maximized which is important to ensure good precision in gaze estimation. It is important to note that the use of a narrow field of view camera is not a pre-requisite of any of the **CR** based methods (including the **CR-DD** and the **PL-CR** methods). The geometrical concept behind all **CR** based methods are independent of the camera setup and, for the **PL-CR** method in particular, this is also true as long as the conditions for the orthographic camera assumption are met.

One obvious drawback of the camera setup used in this work is that, when the head moves, it is very likely that the eye will fall out of the camera's field of view, requiring the manual reposition of the camera (in order to point to the user's eye) and focus adjustment (since a manual focus camera was used). Although this was the case during the conduction of the user experiments, this limitation can be easily solved by the use of a pan-tilt unit in conjunction with an auto focus camera. Another alternative to overcome this limitation is the use of high resolution cameras. Using higher resolutions it is possible to keep the eye within the field of view during head movement and, at the same time, ensure that the eye region in the captured image has a reasonable size.

An experiment control software was developed for data acquisition. The software was responsible for displaying a circular target at each of the 49 test points on screen and storing the image of the subject's eye. Starting from the top left point among the 49 test points, the target was displayed in a left to right and top to bottom sequence. At each test position the target stayed for about 1.3 seconds (equivalent to 40 video frames). During this time 20 images of the subject's eye were stored. Also, during this interval, the size of the circular target varied from an initial radius of 20 pixels to a final radius of 5 pixels to serve as visual stimulus. Since multiple samples were used for each test point, the gaze estimation error for a given target point was computed as the average gaze estimation error for all samples for that target. A chin rest fixed to a tripod was used to maintain the subjects' head still during image acquisition at each head position.

Besides subjects' participation, the data acquisition involved the participation of an operator responsible for controlling the software. The data acquisition process for each subject followed the protocol:

1. the operator explains the objectives of the experiment, how data acquisition is done, and gives the subject the consent form;
2. the operator places the chin rest at position P_0;
3. the subject sits down and accommodates his/her head on the chin rest;
4. the operator directs the camera to the subject's right eye;
5. the operator adjusts the camera focus;
6. the operator adjusts the parameters from the capture software (thresholds used in image processing);
7. the operator starts the capture process;
8. the subject follows the target that scans through the 49 test points;
9. steps 4 to 8 are repeated for the subject's left eye;
10. the operator places the chin rest in the next head position and steps 3 to 9 are repeated for all head positions.

Note that data acquisition was carried out using binocular vision. Both right and left eyes had a clear vision of the screen, regardless of which eye images were being captured for.

6.3.1 Outlier Filtering

Incomplete feature vectors (where one of the expected corneal reflections or the pupil are missing) and incorrect feature detection (where points that do not correspond to the reflections or the pupil are detected as if they were) were problems faced during processing of captured data. A manual evaluation of detected features would be impractical due to the large amount of collected images (20 samples per point × 49 test points × 9 head positions). For this reason an automated approach was used to perform this evaluation, in which all feature vectors resulting from feature detection were classified as being either good or bad. Just the good feature vectors were used to evaluate the performance of the gaze estimation methods being tested.

To do such evaluation of feature vectors, we analyzed the whole set of feature vectors (i.e., all samples for all test points) for a given head position. The analysis was carried in two steps: first feature vectors presenting incorrect corneal reflections were discarded and after that feature vectors containing bad pupil detection were discarded as well.

To discover which corneal reflections were incorrect the following approach was taken: using g_0 as a reference point, and considering all 49 × 20 samples for a given head position, a representative point for g_1, g_2, g_3 and g_4 were computed. It is not expected that each g_i sample matches its representative point, but it is expected that the set of all g_i samples are clustered around its representative. Feature vectors containing at least one g_i whose distance to its representative exceeds a given threshold were classified as bad and discarded.

For pupil filtering a similar approach was taken, but considering just the 20 samples for a given test point. Since the subjects were supposed to be staring at a single point during collection of all 20 samples, it is also expected that the detected pupils are clustered around a representative point. Again a distance criteria was used to classify the pupil detection as either good or bad, and feature vectors containing bad pupils were discarded. In addition to the cases where the pupil was not correctly detected, this pupil filtering was also useful to filter the cases where the subject moved his/her eye ahead of the target point due to prediction.

For both the corneal reflection as well as the pupil filtering, the representative point R for a set of $Q_j = (q_{j_x}, q_{j_y})$ points was computed as:

$$R = \big(median(q_{j_x}), median(q_{j_y})\big) \tag{48}$$

The use of the median to determine the representative of a set of points was motivated by the fact that incorrect detected features usually displays a large displacement from the expected location which would affect computation of a representative based on mean values.

6.4 Experimental Results

A group of 7 subjects participated in the user experiment: 3 females and 4 males with ages ranging from 25 to 65 years. From the 7 subjects, 4 of them (subjects 2, 3, 4 and 7) make daily use of corrective lenses but all of them were capable to visualize the screen targets used during data acquisition with naked eye. The dominant eye for subjects 1 to 7 are, respectively: right, left, right, right, right, left and right. Gaze estimation results for each subject are shown in Figs. 16 through 22.

Similarly to the simulations, it is also expected that the HMC methods exhibit better performance than the non-HMC methods, as the head moves away from the calibration position P_0. These expectations were met for subjects 1, 2, 4, 5, 6 and 7. For these subjects, the HMC methods achieved smaller gaze estimation errors than the non-HMC methods for most head positions. Besides that, under head movement, the error for HMC methods grew at a lower rate when compared to non-HMC methods. Some particular observations regarding the results for subjects 6 and 7 can be made though.

Results for subject 6's right eye are quite reasonable with the exception of test position P_6. For the left eye of subject 6, results show that although the HMC methods perform better then non-HMC methods, results are relatively similar and head movement affects all methods equally. For this case, observe that the size of the displacement vector obtained by calibration of the **CR-D** method is relatively small compared to the size for the right eye and for other subjects as can be seen in Fig. 23. This indicates a smaller κ, as predicted in Sect. 3.1.

For subject 7's right eye, HMC methods do not achieve a smaller error than non-HMC methods at some positions, which is contrary to our expectations. For the left eye, however, the result meets the expectation, showing a clear distinction between the two groups of methods.

Subject 3 illustrates the only case for which a clear distinction in performance cannot be observed when the HMC and non-HMC methods are compared. Also, gaze estimation error of non-HMC methods are not affected by head movements as expected. For example, for this subject's right eye, the gaze estimation error observed at positions P_0 and P_8 are very similar, despite the fact of position P_8 being the farthest from P_0. As for the left eye, inspection of the displacement vector length (see Fig. 23) also suggests a small κ as the reason why the results are not the expected.

In general, experimental results are consistent with the ocular dominance for the subjects, in the sense that results obtained for the dominant eye are in accordance with the expectations, the exceptions being subjects 3 and 7. The unclear results for subject 3 makes it difficult to analyze the results considering ocular dominance. For subject 7, results for the left (non-dominant) eye are superior than the results for the right (dominant) eye. Results for subject 7's left eye show a clear distinction in performance between the HMC and non-HMC methods that is not observed for the right eye.

One reason that explains these variation of results between different subjects is related to the κ angle. When a subject's κ is small, the effects of head movements for non-HMC methods are smaller and consequently the new proposed methods do not show a significant accuracy improvement when the subject moves his/her head. The magnitude of κ can be inferred by the size of the displacement vector used by the **CR-D** method. Figure 23 shows the size of the displacement vectors obtained by calibration of the **CR-D** method, for all 7 subjects. It is possible to see that the left eyes of subjects 3 and 6 have the smallest κ and some of the lowest improvement ratios for the **CR-DD** and **PL-CR** methods (when compared to the **CR-D** and **HOM** methods).

Note that even in cases where improvements for the HMC methods are absent, the performance of the HMC methods is, in the worst case, equal to the performance observed for the non-HMC methods. HMC methods can, therefore, be used for any subject, even for subjects for whom the expected gaze estimation improvement is not significant.

The combined result for all 7 subjects is shown in Fig. 24 as well as on Table 1. Figure 24 shows the average gaze estimation error (in degrees) for all subjects, at each head position. In addition to the average estimation error, Table 1 also presents the observed standard deviation for all subjects. In this table, results for the *Pupil-Corneal Reflection* (**PCR**) method are also included for comparison purposes. Results for the **PCR** method were not included in any of the Figs. 16 to 22 and Fig. 24 since the high estimation errors for this method would make it difficult to compare the performance among the different **CR** based methods being evaluated. Note that the **PCR** method exhibit the best result among all methods at position P_0, but is no match for the **CR** based methods (even the non-HMC ones) when the head moves away from the calibration position.

Results from Fig. 24 and Table 1 show that, on average, HMC methods achieve lower error than non-HMC methods. In addition to that, the average error for the HMC methods tends to be more stable across all head positions when compared to the non-HMC methods. Average error ranges from

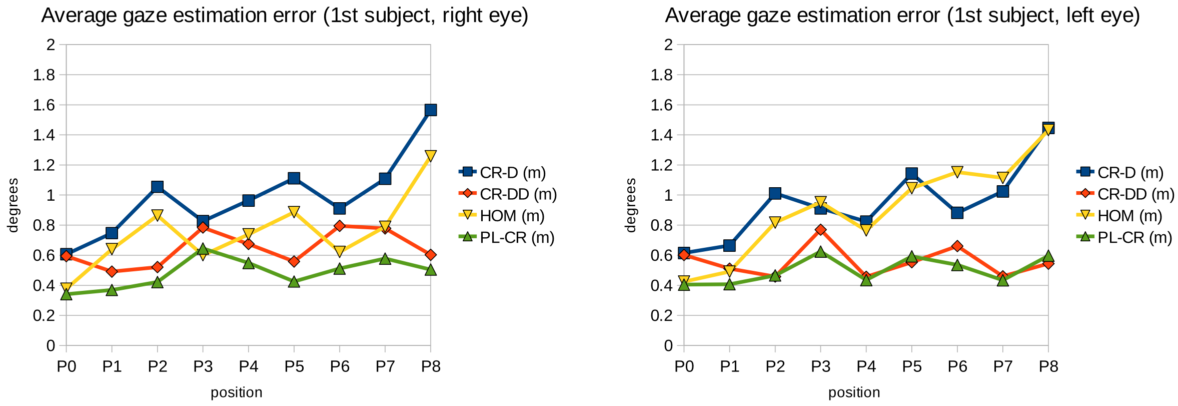


Fig. 16 Average gaze estimation error (measured in degrees of visual angle), for left and right eyes of 1st subject, at test positions P_0 through P_8. This subject is right-eye dominant

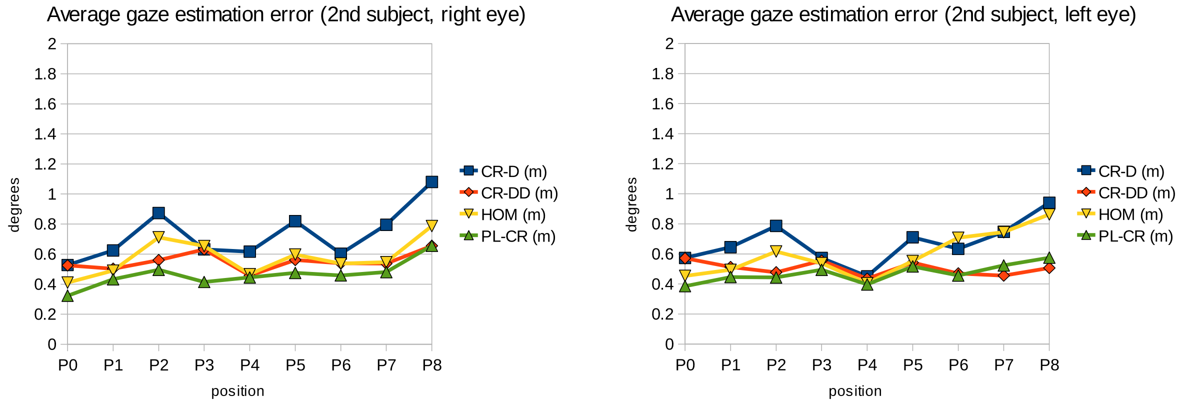


Fig. 17 Average gaze estimation error (measured in degrees of visual angle), for left and right eyes of 2nd subject, at test positions P_0 through P_8. This subject is left-eye dominant

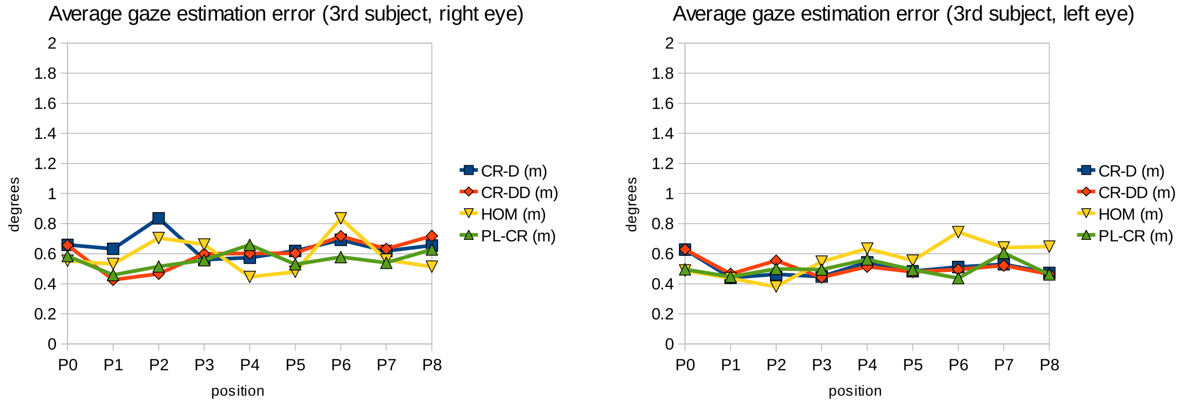


Fig. 18 Average gaze estimation error (measured in degrees of visual angle), for left and right eyes of 3rd subject, at test positions P_0 through P_8. This subject is right-eye dominant

0.49° to 0.62° for the **CR-DD** method, 0.38° to 0.59° for the **PL-CR** method, 0.56° to 1.01° for the **CR-D** method and 0.44° to 0.93° for the **HOM** method. Note also that the standard deviation for the HMC methods is approximately the same for all head positions, in contrast to the standard deviation for the non-HMC methods that tends to grow as the head moves away from P_0. Standard deviation ranges from 0.21° to 0.28° for the **CR-DD** method, 0.18° to 0.22°

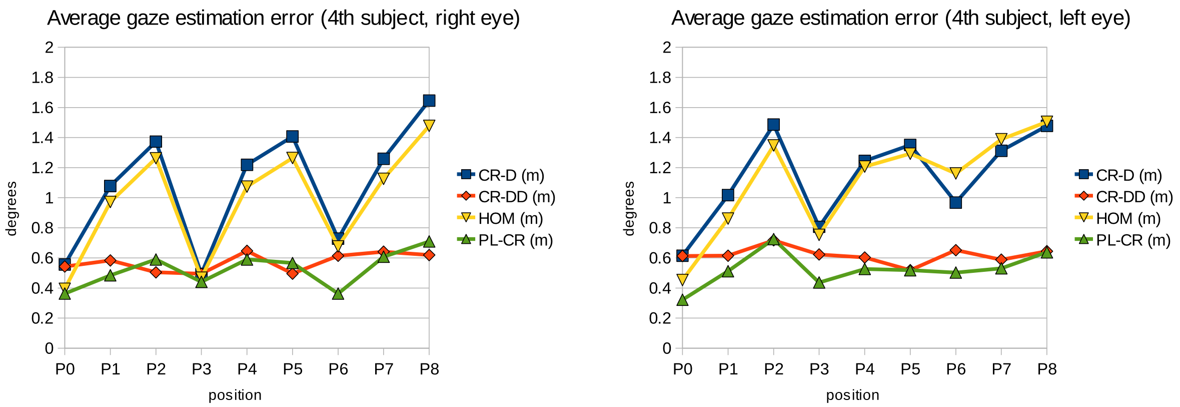


Fig. 19 Average gaze estimation error (measured in degrees of visual angle), for left and right eyes of 4th subject, at test positions P_0 through P_8. This subject is right-eye dominant

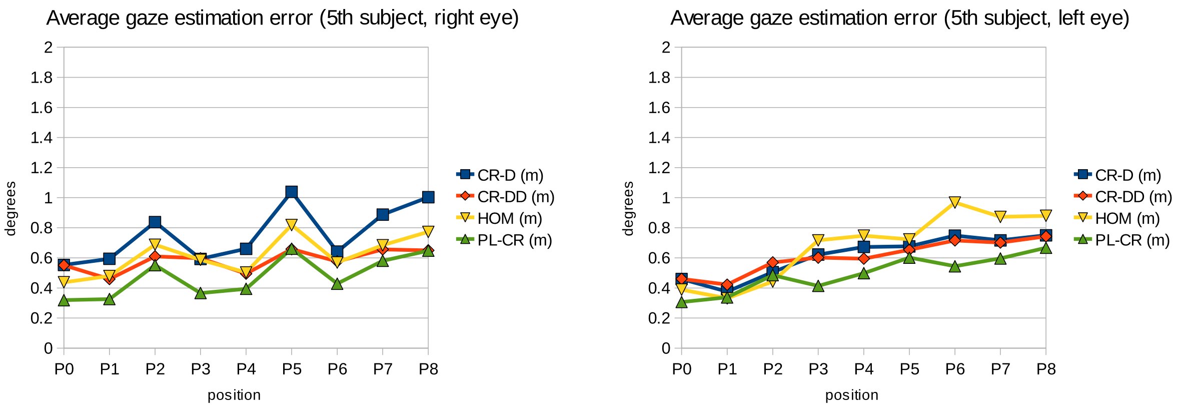


Fig. 20 Average gaze estimation error (measured in degrees of visual angle), for left and right eyes of 5th subject, at test positions P_0 through P_8. This subject is right-eye dominant

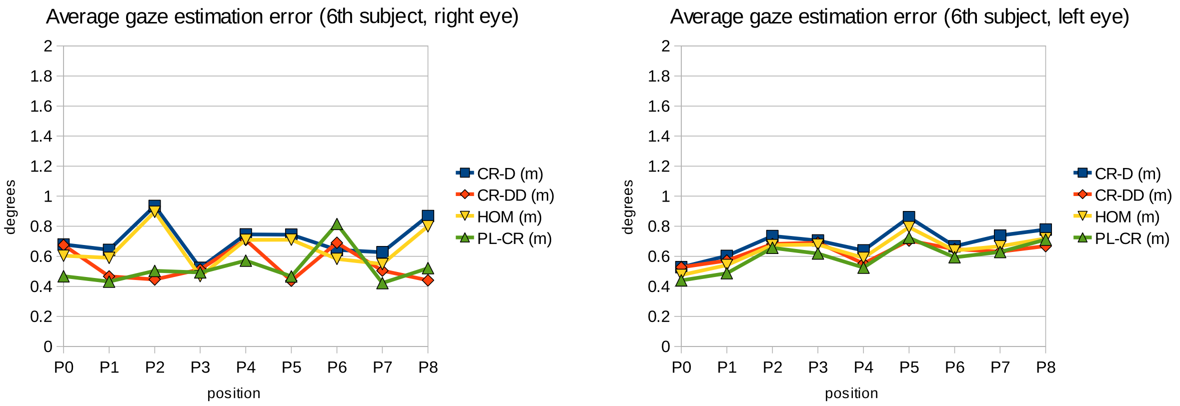


Fig. 21 Average gaze estimation error (measured in degrees of visual angle), for left and right eyes of 6th subject, at test positions P_0 through P_8. This subject is left-eye dominant

for the **PL-CR** method, 0.26° to 0.43° for the **CR-D** method and 0.23° to 0.41° for the **HOM** method. As expected, the improvements of the head movement compensation strategies employed by the **CR-DD** and **PL-CR** methods are more noticeable at head positions that are farther from P_0. In these cases, average reductions of up to 40 % in the gaze estimation error are achieved.

Considering just the non-HMC methods, the experimental results show a better performance of the **HOM** method over the **CR-D** method. This result is expected since the **CR-**

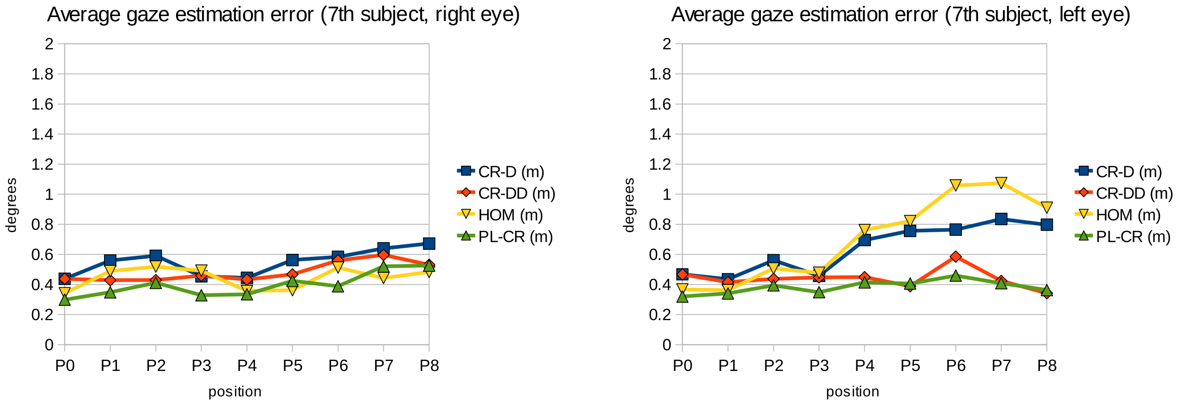


Fig. 22 Average gaze estimation error (measured in degrees of visual angle), for left and right eyes of 7th subject, at test positions P_0 through P_8. This subject is right-eye dominant

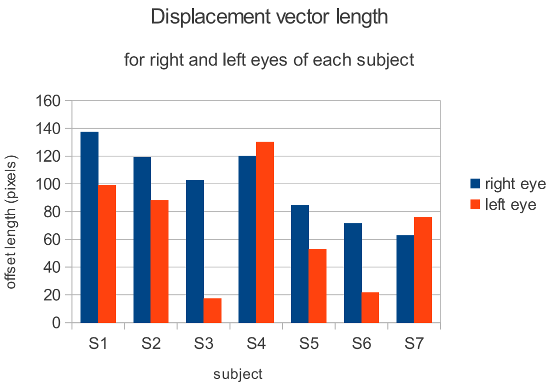


Fig. 23 Size of displacement vector used by methods CR-D and CR-DD, obtained after calibration at P_0

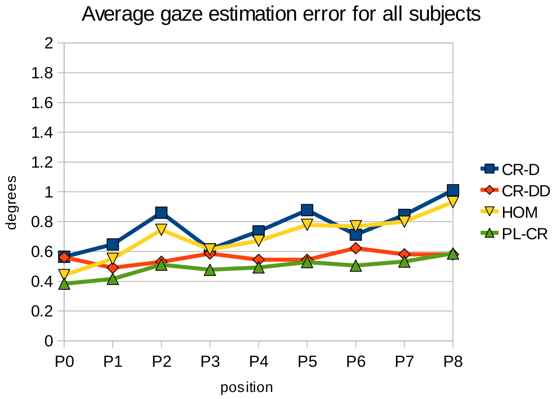


Fig. 24 Average gaze estimation error for all users, at test positions P_0 through P_8

D employs just two basic transformations to correct the PoR estimation: a scale (of the pupil center) and a translation (addition of the displacement vector). The displacement vector is obtained so that gaze estimation error over the entire screen is minimized on average, but is not optimal for individual screen points. The **HOM** method, in contrast, uses a homography normalization to compensate sources of error of the **CR** method. The homography transformation has more degrees of freedom, being more flexible and resulting in better correction for each individual screen point. This means a smaller gaze estimation error for individual points which results in a lower average gaze estimation error than the **CR-D** method.

As for the HMC methods, the experimental results show that the **PL-CR** method performs better than the **CR-DD** method, mainly for positions where the head is closer to the screen. As the head gets farther from the screen the performance difference between these two methods gets smaller until the observed gaze estimation errors are very similar. This is explained by the **CR-DD** method's approach to compensate head movements that just considers distance variation, ignoring eye rotation. Because of this, at distances closer to the screen (where the eye needs to rotate more), the head movement compensation is not as affective as the one employed by the **PL-CR** method.

All these observations related to the experimental results for both the HMC and non-HMC methods are consistent with the simulation results previously presented.

Since the HMC methods rely on extra computation to compensate head movements, inaccuracies of detected feature points will be propagated to the head compensation schemes used by each. An important note about the **CR-DD** method in particular is that the method considers distance variation relative to the screen, but in fact what is being computed is distance variation relative to the camera. In the user experiments the camera was positioned about 10 cm in front of the screen. This difference may introduce some error in the estimation of distance variation. Potential sources of error for the **PL-CR** includes eventual discrepancies between the eye model used and subject's eyes. It is known for example that the radius of the cornea surface changes from its central region to its borders (Nagamatsu et al. 2010a).

Table 1 Average gaze estimation error and standard deviation for all subjects (in degrees of visual angle). Each line contains results for a head position and each column corresponds to a gaze estimation method

	PCR	CR-D	CR-DD	HOM	PL-CR
P0	0.35 ± 0.23	0.56 ± 0.26	0.56 ± 0.26	0.44 ± 0.23	0.38 ± 0.21
P1	6.07 ± 2.08	0.65 ± 0.28	0.49 ± 0.21	0.55 ± 0.25	0.42 ± 0.18
P2	7.93 ± 2.72	0.86 ± 0.36	0.53 ± 0.21	0.75 ± 0.34	0.51 ± 0.20
P3	2.52 ± 1.06	0.61 ± 0.29	0.59 ± 0.28	0.61 ± 0.29	0.48 ± 0.22
P4	6.65 ± 2.27	0.73 ± 0.34	0.54 ± 0.22	0.67 ± 0.32	0.49 ± 0.21
P5	8.18 ± 2.79	0.88 ± 0.35	0.54 ± 0.21	0.78 ± 0.34	0.53 ± 0.21
P6	4.80 ± 1.61	0.71 ± 0.29	0.62 ± 0.26	0.77 ± 0.34	0.51 ± 0.22
P7	7.43 ± 2.50	0.85 ± 0.34	0.58 ± 0.23	0.80 ± 0.36	0.53 ± 0.20
P8	8.31 ± 2.83	1.01 ± 0.43	0.58 ± 0.23	0.93 ± 0.41	0.59 ± 0.21

Eye torsion was also ignored for both HMC methods. Although Guestrin and Eizenman (2010) argued that practical effect of this kind of eye movement (considering distances of 60–70 cm from the screen) in gaze estimation results are very small, it would be interesting to further investigate and consider this kind of movement in the model in a future work.

7 Implementation

Our implementation was constrained by the hardware that we had available in our lab. Our gaze tracking device consists of an analog monochrome video camera, a USB video capture card, a desktop computer, and several light sources that are required by the cross-ratio methods.

Light sources consist of infrared LEDs and are divided into two sets. One set correspond to the L_i points and are attached around the monitor. They generate corneal reflections G_i that are imaged as g_i. The second set is attached around the camera's optical axis. This set generate the reference corneal reflection G_0 imaged as g_0. A filter is also used in the camera to filter light in the visible spectrum.

A circuit that process the analog video signal controls activation of these two sets of LEDs. While the even field is scanned, the first set of LEDs (screen) is activated. Besides generation of g_i, the pupil appears dark in the even lines of the image (as we usually see it). While the odd field is scanned, the second set of LEDs (camera) is activated. Besides generation of g_0, light is reflected from the back of eye, and the pupil appears bright at the odd lines of the image.

Since the pupil is usually the only image element that exhibits a large contrast between the two illumination conditions, this alternating illumination scheme facilitates pupil detection during image processing.

7.1 Software

The gaze tracking software was developed for the Linux platform and uses the OpenCV library for image processing. It works in both real time and offline modes, and implements the **PCR**, **CR-D**, **CR-DD**, **HOM**, and **PL-CR** methods for remote eye gaze tracking. The gaze tracking software is responsible for video acquisition (when operating in real time mode), image processing, and detection of eye features which are then passed to the implementations of the gaze estimation methods. Image processing and feature detection mainly consist of pupil and corneal reflection detection. For the **PL-CR** method, in particular, iris detection is also performed.

Pupil detection is based on the differential method (Ebisawa 1995). The first step consists in deinterlacing of the input image, producing a *bright* and a *dark* pupil image, followed by subtraction of the *dark* image from the *bright* one. The resulting image, *diff*, is then thresholded to segment its high contrast regions, resulting in *diffT*. To avoid considering very bright regions (corneal reflections or specular reflections over glasses) as pupil candidates, or as part of the pupil, the brightest areas of both *dark* and *bright* images are segmented by two more threshold operations, resulting in *darkB* and *brightB*. To make pupil detection more robust, a threshold is also applied to the *dark* image to select its darkest regions, resulting in *darkT*. A binary image containing regions that are pupil candidates (i.e., present high contrast between *dark* and *bright* images, appear as dark regions in the *dark* image and are not extremely bright in any of the *dark* and *bright* images) is given as the result of following boolean operations:

$$candidate = diffT \wedge darkT \wedge (!darkB) \wedge (!brightB) \qquad (49)$$

After the *candidate* image is computed, connected component regions are extracted and analyzed to select one as the best pupil candidate. Ideally we expect to have just one candidate blob but, in some situations, especially for people wearing glasses, it is possible to have more than one. The best candidate is selected as largest blob that satisfies some conditions: the aspect ratio of the bounding box around the blob must be ≥ 0.5 and ≥ 2.0, and also the fill ratio (area of the blob divided by the area of the bounding rectangle) must be >0.5 (just for comparison the fill rate of a circular shape is approximately 0.785).

After the best candidate is selected, its contour is extracted into the *contour* image. Care must be taken here, because if a corneal reflection is formed over the pupil edge, the pupil contour will be corrupted by part of the reflection contour. We eliminate this interference by dilating *darkB* and *brightB* and subtracting both of them from the *contour* image. The resulting pixels in the contour are then used to fit an ellipse, that is taken as the pupil, and p is taken as the ellipse center.

Since corneal reflections appear as bright spots in the images of the eye, we detect them by segmenting bright regions in both the *dark* and *bright* images. In fact this step is executed for pupil detection, resulting in *darkB* and *brightB*. The remaining blobs in both *darkB* and *brightB* are ordered according to their distances from the pupil center previously computed. In *brightB* we expect to find one corneal reflection, so the closest blob to the pupil center is taken as g_0. For the *darkB* image, the 5 closest blobs to the pupil center are selected and combinations of 4 blobs are tested against a rectangularity criteria. Given 4 points that form a quadrilateral, and its internal angles $\hat{a}_j$, the rectangularity is the sum of $|\hat{a}_j - 90|$, for $j \in \{1, 2, 3, 4\}$. The smaller this sum is, the closest the quadrilateral is to a rectangle. The combination that has the smallest rectangularity score is taken as the set of corneal reflections g_i.

For the **PL-CR** method, in addition to the pupil and corneal reflections, the iris must also be detected. Instead of actually detecting the iris contour, we compute an approximation of it based on the detected pupil. It is assumed that the iris is imaged as an ellipse with the same center and shape of the pupil. The iris approximation is computed by scaling the detected pupil so that the resulting ellipse's contour best matches the contour of the actual iris.

The computation of the approximate iris based on the scale of the pupil eliminates the need to detect the full iris contour, but at least one point from the contour must be detected in order to determine by how much the pupil needs to be scaled. To increase the chance of successful detection of such contour point, we look for it in the horizontal line passing through p. This search starts at p and follows in the direction of the corneal reflection g_0. This strategy ensures that the iris contour point belonging to the horizontal search line will not be occluded by eyelids or eye corners.

The gradient vector $\mathbf{gr}$ of an arbitrary point ic that belongs to the iris contour is expected to have a large magnitude value and to also point in the approximate direction given by $\mathbf{i} = (ic - p)$. If ic belongs to the same horizontal line as p, $\mathbf{i}$ can be expressed by $\mathbf{i} = (1, 0)$ or $\mathbf{i} = (-1, 0)$ depending on the search direction. This way, the point in the horizontal search line that maximizes the following score

$$score = \left(\frac{\mathbf{gr}}{\|\mathbf{gr}\|} \cdot \frac{\mathbf{i}}{\|\mathbf{i}\|} \right) \|\mathbf{gr}\| \tag{50}$$

is taken as a point belonging to the iris contour, which will then be used to scale the pupil.

The use of an iris approximation is reasonable since the iris normal computation does not rely on its contour but rather on image points p and g_0, and the c_z model parameter. The iris contour is just used by the **PL-CR** method for computation of the s scale factor as described in Sect. 5.3.

Our gaze tracker implementation is able to process each video frame in approximately 12 milliseconds using one core of a Xeon 2.8 GHz processor, which ensures the real time operation, an essential requirement when we have interactive applications in mind. This processing time was achieved for the **PL-CR** method, which is the one that demands more computation to estimate the gaze. This time also includes the time spent during the display of the captured video.

8 Conclusion

In this paper we presented two new methods for remote eye gaze tracking developed with the objective of improving head movement tolerance of current cross-ratio based eye trackers: the **CR-DD** and the **PL-CR** methods.

The **CR-DD** method is an extension of the **CR-D** method in which the size of the displacement vector is adjusted dynamically according to the eye distance from the screen. Instead of absolutely computing the eye distance, we compute the eye distance variation relative to an initial eye position. This computation is done based on the observed size of the corneal reflection pattern. A problem with this approach is that what we are truly measuring is the distance variation from the camera, but we are taking it as the variation between the eye and screen. It may be acceptable depending on the placement of the camera, but this compensation is not 100 % effective. Another problem is that we are just measuring variation in eye distance, but not eye rotation that also affects the displacement vector.

The **PL-CR** method compensates both sources of errors pointed by Guestrin et al. (2008) by estimating the average Π_G plane where corneal G_i reflections are formed (or the plane over which L_i are projected, having the cornea center as projection center) and computing the intersection of the visual axis of the eye with this plane (V'). Once G_i and V' lies on a common plane and the true visual axis of the eye is being considered, the basic principle of the cross-ratio method can be applied directly. To keep the use of a single non-calibrated camera an orthographic camera model was used to compute Π_G and V'. We also used an eye model whose parameters (obtained via calibration) are invariant regardless of the eye location and orientation. In contrast to the **CR-DD** method, the approach taken for the **PL-CR** better handles all kind of head movements since eye rotation is handled naturally by the computation of the visual axis intersection with Π_G.

These methods were evaluated and compared to the **CR-D** and **HOM** cross-ratio based methods. Both simulations and user experiments were conducted and results confirmed the improvement in gaze estimation accuracy for the two proposed methods under the condition of head motion. We also showed that the amount of observed improvement is dependent on the magnitude of the angle between the visual and optical axes of the eye.

In addition to proposing the **CR-DD** and **PL-CR** methods, we also implemented a gaze tracker that is able to estimate eye gaze in real time. The combination of the remote gaze tracking setup, the proposed new methods that allow larger head movement, and the real time implementation of these methods constitutes an important foundation when the objective is to develop gaze based interactive applications.

Acknowledgements The authors would like to thank Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES), Fundação de Amparo à Pesquisa do Estado de São Paulo (FAPESP) and IBM for their financial support.

References

Cerrolaza, J. J., Villanueva, A., & Cabeza, R. (2008). Taxonomic study of polynomial regressions applied to the calibration of video-oculographic systems. In *Proceedings of the 2008 symposium on eye tracking research & applications, ETRA '08* (pp. 259–266). New York: ACM Press.

Chen, J., Tong, Y., Gray, W., & Ji, Q. (2008). A robust 3d eye gaze tracking system using noise reduction. In *ETRA '08: proceedings of the 2008 symposium on eye tracking research & applications* (pp. 189–196). New York: ACM Press.

Coutinho, F., & Morimoto, C. (2006). Free head motion eye gaze tracking using a single camera and multiple light sources. In *19th Brazilian symposium on computer graphics and image processing, 2006, SIBGRAPI '06* (pp. 171–178).

Duchowski, A. T. (2003). *Eye tracking methodology: theory and practice*. Berlin: Springer.

Ebisawa, Y. (1995). Unconstrained pupil detection technique using two light sources and the image difference method. In *Visualization and intelligent design in engineering and architecture* (pp. 79–89.

Emanuele Trucco, A. V. (1998). *Introductory techniques for 3-D computer vision*. New York: Prentice Hall.

Guestrin, E., & Eizenman, M. (2006). General theory of remote gaze estimation using the pupil center and corneal reflections. *IEEE Transactions on Biomedical Engineering*, *53*(6), 1124–1133.

Guestrin, E. D., & Eizenman, M. (2008). Remote point-of-gaze estimation requiring a single-point calibration for applications with infants. In *ETRA '08: proceedings of the 2008 symposium on eye tracking research & applications* (pp. 267–274). New York: ACM Press.

Guestrin, E. D., & Eizenman, M. (2010). Listing's and Donders' laws and the estimation of the point-of-gaze. In *Proceedings of the 2010 symposium on eye-tracking research & applications, ETRA '10* (pp. 199–202). New York: ACM Press.

Guestrin, E. D., Eizenman, M., Kang, J. J., & Eizenman, E. (2008). Analysis of subject-dependent point-of-gaze estimation bias in the cross-ratios method. In *Proceedings of the 2008 symposium on eye tracking research & applications, ETRA '08* (pp. 237–244). New York: ACM Press.

Hansen, D. W., Agustin, J. S., & Villanueva, A. (2010). Homography normalization for robust gaze estimation in uncalibrated setups. In *Proceedings of the 2010 symposium on eye-tracking research & applications, ETRA '10* (pp. 13–20). New York: ACM Press.

Hansen, D. W., & Ji, Q. (2010). In the eye of the beholder: A survey of models for eyes and gaze. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *32*, 478–500.

Hartley, R., & Zisserman, A. (2000). *Multiple view geometry in computer vision*. Cambridge: Cambridge University Press. ISBN: 0521623049.

Hennessey, C., Noureddin, B., & Lawrence, P. (2006). A single camera eye-gaze tracking system with free head motion. In *Proc. of the ETRA 2006* (pp. 87–94). San Diego, CA.

Kang, J. J., Guestrin, E. D., Maclean, W. J., & Eizenman, M. (2007). Simplifying the cross-ratios method of point-of-gaze estimation. In *30th Canadian medical and biological engineering conference (CMBEC30)*.

Kaufman, A., Bandopadhay, A., & Shaviv, B. (1993). An eye tracking computer user interface. In *Proc. of the research frontier in virtual reality workshop* (pp. 78–84).

Model, D., & Eizenman, M. (2010). User-calibration-free remote gaze estimation system. In *Proceedings of the 2010 symposium on eye-tracking research & applications, ETRA '10* (pp. 29–36). New York: ACM Press.

Morimoto, C. H., & Mimica, M. R. M. (2005). Eye gaze tracking techniques for interactive applications. *Computer Vision and Image Understanding*, *98*(1), 4–24.

Morimoto, C. H., Koons, D., Amir, A., & Flickner, M. (1999). Frame-rate pupil detector and gaze tracker. In *Proceedings of the IEEE ICCV'99 frame-rate computer vision applications workshop*.

Nagamatsu, T., Kamahara, J., Iko, T., & Tanaka, N. (2008). One-point calibration gaze tracking based on eyeball kinematics using stereo cameras. In *ETRA'08* (pp. 95–98).

Nagamatsu, T., Iwamoto, Y., Kamahara, J., Tanaka, N., & Yamamoto, M. (2010a). Gaze estimation method based on an aspherical model of the cornea: surface of revolution about the optical axis of the eye. In *Proceedings of the 2010 symposium on eye-tracking research & applications, ETRA '10* (pp. 255–258). New York: ACM Press.

Nagamatsu, T., Sugano, R., Iwamoto, Y., Kamahara, J., & Tanaka, N. (2010b). User-calibration-free gaze tracking with estimation of the horizontal angles between the visual and the optical axes of both eyes. In *Proceedings of the 2010 symposium on eye-tracking research & applications, ETRA '10* (pp. 251–254). New York: ACM Press.

Robinson, D. A. (1963). A method of measuring eye movements using a scleral search coil in a magnetic field. *IEEE Transactions on Biomedical Engineering*, *10*, 137–145.

Shih, S., & Liu, J. (2003). A novel approach to 3d gaze tracking using stereo cameras. *IEEE Transactions on Systems, Man and Cybernetics Part B Cybernetics*, 1–12 (2003).

Villanueva, A., Cabeza, R., Porta, S., Bohme, M., Droege, D., & Mulvey, F. (2008) Report on new approaches to eye tracking, summary of new algorithms. In *Communication by gaze interaction (COGAIN)*, IST-2003-511598: Deliverable 5.6 (2008).

Wang, J. G., & Sung, E. (2002). Study on eye gaze estimation. *IEEE Transactions on Systems, Man and Cybernetics Part B Cybernetics*, *32*(3), 332–350.

Yoo, D. H., & Chung, M. J. (2005). A novel non-intrusive eye gaze estimation using cross-ratio under large head motion. *Computer Vision and Image Understanding*, *98*(1), 25–51, Special Issue on Eye Detection and Tracking.

Yoo, D. H., Kim, J. H., Lee, B. R., & Chung, M. J. (2002). Non-contact eye gaze tracking system by mapping of corneal reflections. In *Proceedings fifth IEEE international conference on automatic face and gesture recognition, 2002* (pp. 94–99).