IST-2003-511598 (NoE)

COGAIN

Communication by Gaze Interaction

Network of Excellence

Information Society Technologies

D5.2 Report on New Approaches to Eye Tracking

Due date of deliverable: 31.08.2006
Actual submission date: 23.10.2006

Start date of project: 1.9.2004 Duration: 60 months

Siauliai University

Project co-funded by the European Commission within the Sixth Framework Programme (2002-2006)		
Dissemination Level		
PU	Public	x
PP	Restricted to other programme participants (including the Commission Services)	
RE	Restricted to a group specified by the consortium (including the Commission Services)	
CO	Confidential, only for members of the consortium (including the Commission Services)	

Daunys, G. et al. (2006) **D5.2 Report on New Approaches to Eye Tracking**. Communication by Gaze Interaction (COGAIN), IST-2003-511598: Deliverable 5.2. Available at http://www.cogain.org/results/reports/COGAIN-D5.2.pdf

Contributors: Gintautas Daunys (SU)
Bjarne Kjær Ersbøll (DTU)
Martin Böhme (UzL)
Olga Stepankova (CTU)
Arantxa Villanueva (UPNA)
Erhardt Barth (UzL)
Martin Vester-Christensen (DTU)
Tobi Delbruck (UNICH)
Donatas Dervinis (SU)
Detlev Droege (UNI KO-LD)
Jan Fejt (CTU)
Marcela Fejtová (CTU)
Dan Witzner Hansen (ITU)
Lars Kai Hansen (DTU)
Denis Leimberg (DTU)
André Meyer (UzL)
Thomas Martinetz (UzL)
Nerijus Ramanauskas (SU)
Vytautas Vysniauskas (SU)

Table of Contents

Executive Summary

This deliverable is about the algorithms of gaze tracking developed or advanced by COGAIN partners. The main research directions in gaze tracking, chosen by partners, are:

- Remote gaze tracking in infrared light;
- Eye tracking in visible light;
- 3D head pose estimation;
- Gaze tracking with cheap head mounted system;
- Development of silicon retina for gaze tracking;
- Use of Time of Flight cameras.

In the current deliverable the results of first four directions are presented. A mathematical model of video-oculography is introduced. The algorithms for pupil centre and corneal reflections coordinates evaluation are analysed. Advancing of deformable template matching and estimate/minimize algorithms also find place in the deliverable. Algorithms for 3D head pose estimation were proposed. At the end of the deliverable examples of practical implementation of the algorithms into gaze tracking systems are described.

1 Introduction

1.1 Gaze Tracking

The gaze tracker is a key component of gaze communication systems. Gaze tracking has its origin in the research of eye movements. First eye movement recording devices were used in laboratories for investigations in the fields of physiology, ophthalmology, psychology. Progress in eye tracking has enabled us to obtain eye angular positions in real time. The first systems for disabled to control computers by gaze were created. Nowadays there are some commercially available systems (see http://www.cogain.org/eyetrackers) but their expensiveness limits their availability for everyday use. Development of cheap gaze trackers is one of the objectives of the EU Network of Excellence COGAIN.

The desired output from a gaze tracker for gaze interaction is the point of gaze (point of regard). Point of gaze (POG) is the point on a monitor's screen that is imaged on the centre of the highest acuity region (fovea) of the user's eye retina. In other words it is the point of intersection of the eyes visual axis (line of sight) with the plane of the monitor's screen.

The eye as a rigid body has six degrees of freedom. Its location in space is described by three linear coordinates and three angular coordinates. Two more angular coordinates are needed to describe the fovea centre position in the eye. The last two coordinates are constants for the selected eye of the user. During eye tracking we must first track six components. Usually we need only five of the components, because the changes in the third angular co-ordinate of the eye (torsion) are small and it is accepted that it doesn't affect the direction of gaze. Furthermore, this coordinate is dependent on the first two angular coordinates. The five coordinates of the eye together with two constants for the fovea location defines the origin of the line of sight (three linear coordinates) and its orientation (two angular coordinates). We can evaluate point of gaze from the five coordinates, because we assume that monitor position is fixed.

Already, when the survey of Young and Sheena [1] was published, different eye movement measurement techniques were known. The most suitable techniques for computer control are video based methods. The main advantage of these methods is that they are non-invasive. The price for this compared to invasive techniques is lower spatial and time resolution.

A head-mounted eye tracker defines the gaze orientation relative to the head's coordinate system. Other systems track the head position and orientation in space. Even though head-mounted gaze estimation systems are preferred for applications that require large and fast head movements, they have higher intrusion level than remote systems.

Most suitable for computer control are remote eye tracking systems. These systems do not require any equipment to be mounted on the user and allow the user to move their head freely within certain limits. The accuracy of the systems is lower, because of the problem that the user's eye can go out of the visual field of the camera. Therefore, cameras with low zoom or automatic camera orientation mechanisms are needed.

1.2 COGAIN Gaze Tracking Research Directions

The main research directions the COGAIN partners are working on include:

- Remote gaze tracking in infrared light (SU, UzL);
- Eye tracking in visible light (DTU);
- 3D head pose estimation (DTU, SU);

- Gaze tracking with cheap head mounted system (CTU);
- Development of silicon retina for gaze tracking (UNIZH);
- Use of Time of Flight (TOF) cameras (DTU, UzL).

1.2.1 Remote gaze tracking in infrared light

Infrared (IR) lighting allows one to achieve sharp contrast between the eye's iris and pupil regions, so such systems are more accurate than systems with passive lighting. In recent years, a number of so-called remote eye-tracking systems have been described in the literature (see e.g. [2] for a review). The most accurate remote eye tracking systems that have been described in the literature to date use multiple cameras and achieve an accuracy of 0.5 to 1.0 degrees [3–7]. That single-camera systems can achieve accuracy in the range of 0.5 to 1.0 degrees is demonstrated by a commercial system, e.g. [8], but no implementation details have been published.

A new trend here is gaze estimation based on a physical model of the eye. This approach is welcomed in HCI. Most existing gaze tracking systems need to be calibrated before every work session even for frequent use. Image features, such as position of the pupil centre, corneal reflection, limbus, etc. are measured and mapped to screen coordinates using coefficients obtained from calibration data. The burden of calibration is large since users must watch many calibration points, e.g. 5-25 points. Such systems mostly lack robustness to variation of measurement conditions. This approach has been called the "bottom-up" approach [9].

Another approach is model-based (also called the "top-down" approach). Using the second approach the pose parameters of a model of the eye are adjusted in conjunction with a camera model to obtain a match to image data. Recently some works were published about this approach [9, 10, 11]. It seems that such an approach allows one to reduce the calibration effort.

In Section 2 of the deliverable the mathematical model of videooculography, developed by partners of COGAIN, is presented. Section 3 is devoted to algorithms to find pupil centre and corneal reflection coordinates in an image. The Starburst algorithm does not originate from the COGAIN collaboration. However, the analysis of it has been done because its authors from Iowa State University offer an open source implementation called "openEyes" (http://hcvl.hci.iastate.edu/openEyes). Another algorithm – the coordinate averaging algorithm – was created at Siauliai University. Details on the gaze tracking system from the University of Lübeck, which uses the Starburst algorithm, are given in Section 6.1. Details on cheap system from University of Koblenz-Landau are given in Section 6.2.

1.2.2 Eye tracking in visible light

The success of gaze tracking in IR is highly dependent on external light sources and the apparent size of the pupil. Efforts are made to focus on improving eye tracking under various light conditions. Sun light and glasses can seriously disturb the reflective properties of IR light. IR light and synchronization schemes can in general not be exploited when using COTS (Commercial Off-The-Shelf) for eye tracking as IR light emitters cannot be bought off-the-shelf. Attempts to track the eye in natural lighting are described in Section 4.

1.2.3 3D head pose estimation

In order to accurately calculate the point of gaze the 3D pose of the head has to be estimated. Various types of transducers can be used to measure the head pose. In laboratory conditions the magnetic position transducer [12] is the most common. The most appropriate approach for HCI is computer vision. Results of such investigations are presented in Section 5.

1.2.4 Gaze tracking with cheap head mounted system

Revolutionary technological innovations enabled considerable advances in most branches of industry during the last decade. Current producers of electrical components compete by offering improved products with better usability parameters – size being one of them. The companies aim for miniature products, which do not lag behind the classical solutions in any of the relevant functionalities or which even outperform them. This trend also significantly influenced the production of CCD (Charge-Coupled Device) cameras and of various camera modules – new high quality miniature cameras have recently appeared on the market. Their small sizes as well as other very useful properties present them as conducive to innovative applications meeting requirements of various users (see COGAIN Deliverable D3.2[1]) and also to construction of affordable AI solutions [13]

In the context of eye trackers, miniaturisation has made it possible to place a camera within close range of the user's eye. The system called I4Control® was developed at the Czech Technical University. More details about it are found in Section 6.2.

1.2.5 Development of silicon retina for gaze tracking

Some details about this task are given in Section 6.3. More will be presented in a future WP5 deliverable.

1.2.6 Use of Time of Flight cameras

Recent years have seen the development of so-called 3D time-of-flight (TOF) cameras [14]. In addition to providing an intensity image like a conventional camera, these cameras also provide a depth image that gives the distance of the object in the scene at each pixel. This allows the three-dimensional shape of the scene, e.g. the user's head, to be reconstructed. Many recognition and tracking tasks can be implemented more robustly on 3D range data than on intensity images, therefore this technology has the potential to be used for robust head and eye tracking. Two participants in COGAIN, together with other European partners, will be working on TOF-based eye, head and gesture tracking within an EU STREP-project ARTTS[2].

[1] Available online at http://www.cogain.org/results/reports

[2] The project web page is under construction at http://www.artts.eu. Later it will contain information about the project.

2 Mathematical Model of Videooculography

The aim of Section 2 is to create the mathematical model for the eye image in the image sensor formation. The initial data for mathematical modeling are the mutual disposition (placement) of gaze tracking system's components in respect to the subject eye. The model of eye is described in Section 2.1. For gaze tracking the features as eye pupil centre coordinates and corneal reflections, which are caused by light sources, must be evaluated with subpixel accuracy. The coordinates systems, used for simulation of pupil centre and corneal reflection coordinates in image, are described in Section 2.2. The aim of section 2.3 is to investigate theoretically the formation of the eye pupil's image position in the image sensor and define factors, which influence the pupil's centre coordinates. The influence of corneal optics was included in analysis. In the section 2.4 the formation of positions of corneal reflections are simulated. The results of simulation could be used in different algorithms for gaze tracking.

2.1 Eye Model

We used the eye model, depicted in Figure 2.1, for derivation of mathematical model of videooculography. The cornea lies in front of the eye and provides a transparent protective covering. Its refraction index n=1.376. The cornea is occluded by two spherical surfaces. For statistical human eye the curvature radius of external surface R1=7.8 mm, internal surface – R2=6.5 mm. The curvature centres have displacement. This causes different cornea thickness in centre and in periphery. The refraction index of aqueous humour, which fills anterior chamber, is very close to one of cornea (n=1.336). Because refraction indexes of both media are very close, usually all anterior part of eye till iris is described with the same refraction index.

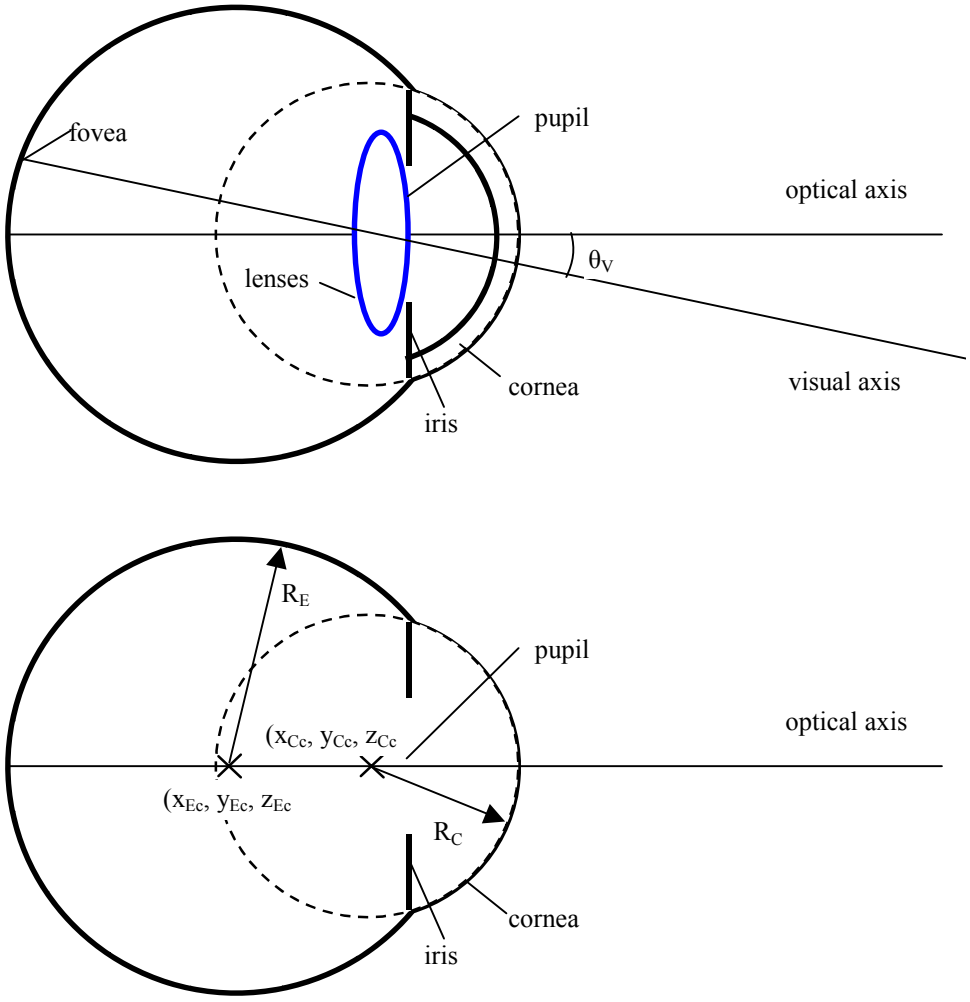


Figure 2.1: (top) Eye model, (bottom) simplified eye model.

The optical axis of eye is the same as of the eye lens. The visual axis is the line defined by the centre of the fovea and the centre of the eye lens. In a typical adult human eye, the fovea falls about 4-5 deg temporally and about 1.5 deg below the point of intersection of the optic axis and the retina.

The following symbols were used in Figure 2.1:

θ_V – angle between visual and optical axis in horizontal plane;

R_E – eye ball radius;

R_C – external cornea curvature radius;

x_{Cc}, y_{Cc}, z_{Cc} – x, y, z co-ordinates of cornea curvature centre;

x_{Ec}, y_{Ec}, z_{Ec} – x, y, z co-ordinates of eye rotation centre.

2.2 Coordinate Systems

Descartes left hand coordinates system was selected. The coordinates system centre is selected near the user's eye. The x axis is selected in direction from user's eye to the monitor screen. The y axis is across horizontal axis and the z axis is across the vertical screen axis (see Figure 2.2 below).

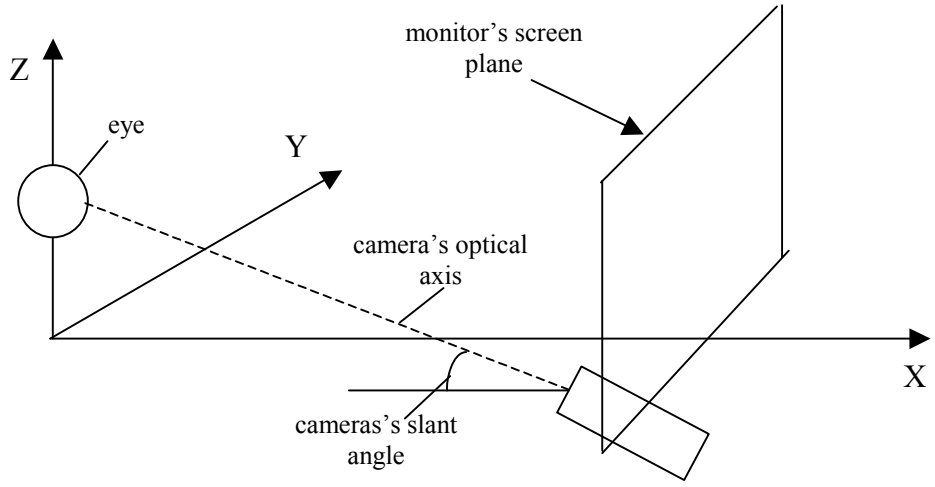


Figure 2.2: (top) Example of physical layout, (bottom) laboratory coordinates system.

Rotation about Z axis is described by angle θ. Rotation about Y axis is described by angle φ. The rotation matrixes can be used for the co-ordinates transformation:

$$\mathbf{A_1} = \begin{pmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{pmatrix}; \quad (2.1)$$

$$\mathbf{A_2} = \begin{pmatrix} \cos\varphi & 0 & \sin\varphi \\ 0 & 1 & 0 \\ -\sin\varphi & 0 & \cos\varphi \end{pmatrix}. \quad (2.2)$$

Resuming result of rotations about two axes can be expressed by matrix product:

$$\mathbf{A} = \mathbf{A_2}\mathbf{A_1} \qquad (2.3)$$

Substituting (2.1), (2.2) into (2.3) we obtain:

$$\mathbf{A} = \begin{pmatrix} \cos\varphi\cos\theta & -\cos\varphi\sin\theta & \sin\varphi \\ \sin\theta & \cos\theta & 0 \\ -\sin\varphi\cos\theta & \sin\varphi\sin\theta & \cos\varphi \end{pmatrix}. \qquad (2.4)$$

To obtain values of the angles is easier due to measuring distances along axis. So we introduce direction coefficients k_y and k_z:

$$k_Y = \frac{\Delta y}{\Delta x}, \qquad (2.5)$$

$$k_z = \frac{\Delta z}{\Delta x}. \qquad (2.6)$$

The relation between direction coefficients (k_y, k_z) and Euler angles (θ, φ) is following:

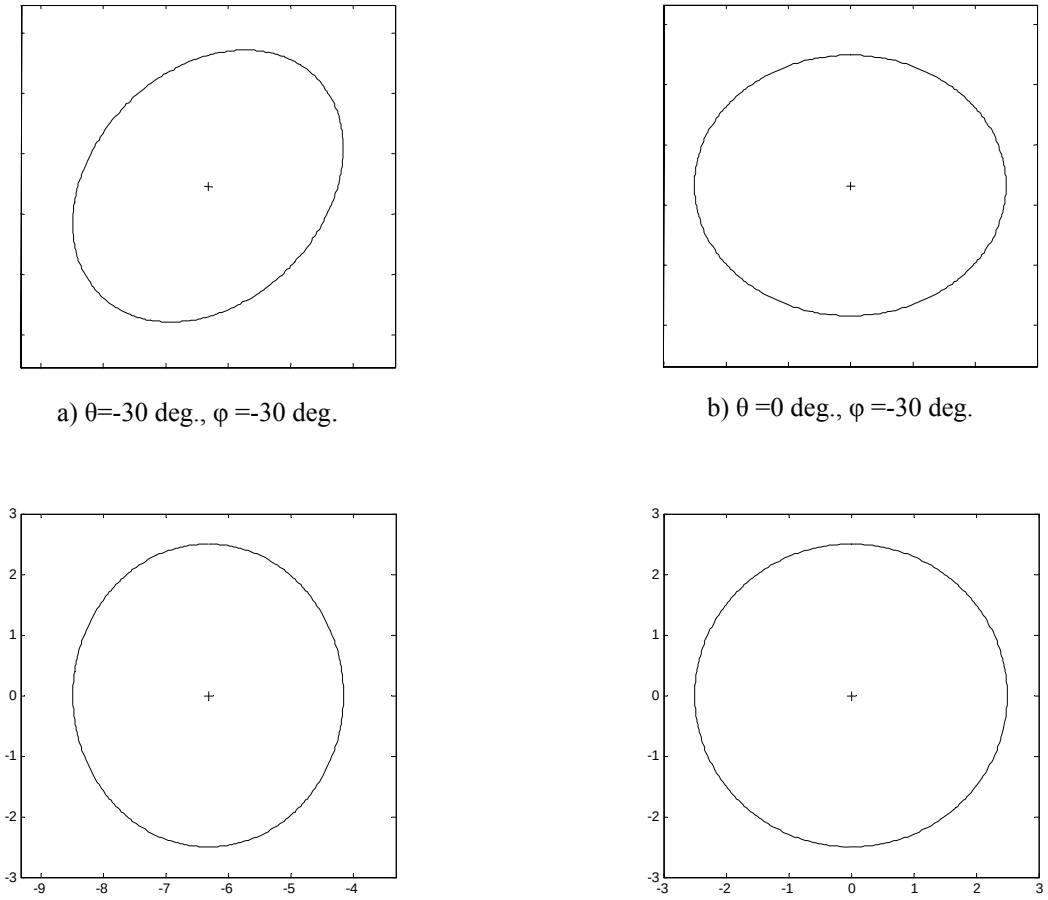
$$\varphi = -arctgk_z; \qquad (2.7)$$

$$\theta = arctg(k_y \cos\varphi). \qquad (2.8)$$

The imaging process in image sensor can be described as transformation from 3D laboratory system to YZ plane of image sensor system.

2.3 Pupil Centre

Pupil centre co-ordinates are often used for finding point of gaze in videooculographical gaze tracking systems. There are some circumstances, which could cause errors of tracking. The pupil is a hole in iris, so its centre could not be tracked directly. Instead, the pupil area or contour are tracked, and from the data, pupil centre coordinates are calculated. We can assume that the pupil form is a circle, when the eye optical axis coincide with the camera optical axis, although the pupil form deviations from a circle were reported [15]. After eye rotation the pupil form changes to an ellipse, which parameters depends on rotation angles (see Figure 2.3). So the pupil form in the frame is a priori unknown.



a) θ=-30 deg., φ =-30 deg.

b) θ =0 deg., φ =-30 deg.

c) θ =-30 deg., φ =0 deg.

d) θ =0 deg., φ =0 deg.

Figure 2.3: Deformation of circle of pupil contour after rotations,
when camera optical axis coincide with the eye optical axis in primary position.

The image of the pupil is obtained through cornea and aqueous humour. The effective index of refraction of the two combined media differs from one of the air and is equal to 1.3375 [16]. The refraction on the surface of the cornea causes that the place of the pupil centre is shifted in the acquired image of eye. Finally, the video camera is usually mounted under the monitor in order to leave free viewing field for the user. The big camera slant angle causes additional iris and pupil image distortions.

Further, we will present the 3D algorithm for modelling light rays propagation. Similar as in recent paper [10, 11] the law of refraction was used on the surface of cornea. The law of refraction states two conditions:

- the incident ray, the refracted ray and the normal at the point of refraction are in the same plane;
- the angle of incidence θ_1, and the angle of refraction θ_2, satisfy Snell's law ($n_1 \sin\theta_1 = n_2 \sin\theta_2$).

In Figure 2.4 the rays propagation from iris to image sensor is presented. The algorithm seeks for such a ray direction from the iris that the refracted ray would become parallel to the optical axis of the video camera. Then distances between rays' co-ordinates at the camera lens plane are proportional to the ones in the image.

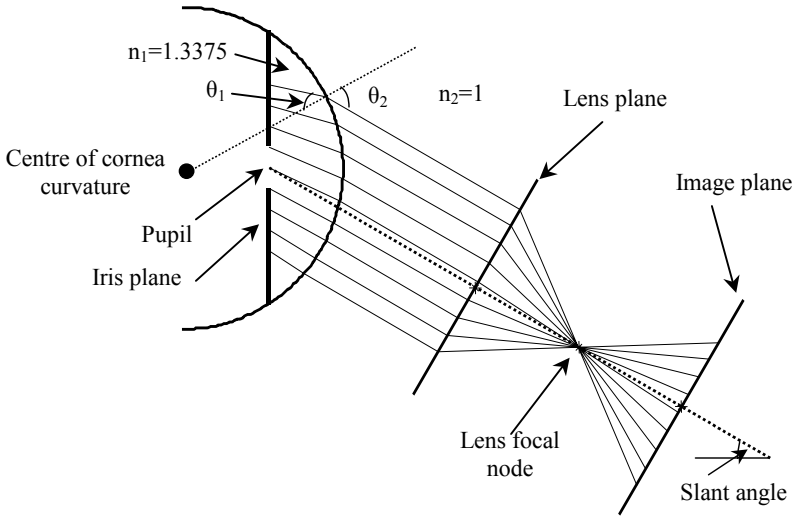


Figure 2.4: Rays propagation from iris to image sensor.

Eye rotation in space can be described by rotation matrix of form (2.4):

$$\mathbf{A}_E = \begin{pmatrix} \cos\varphi_E \cos\theta_E & -\cos\varphi_E \sin\theta_E & \sin\varphi_E \\ \sin\theta_E & \cos\theta_E & 0 \\ -\sin\varphi_E \cos\theta_E & \sin\varphi_E \sin\theta_E & \cos\varphi_E \end{pmatrix} \quad (2.9)$$

where θ_E , φ_E – eye rotation Euler angles.

The initial position of eye points can be described by 3D vectors in the eye coordinate system, where the centre of coordinates system is the eye rotation centre. After rotation, the eye rotation centre remains in the old position when all other points change their position. The initial position of cornea curvature centre is described by vector:

$$\mathbf{r}_{\mathbf{Cc0}} = \begin{pmatrix} x_{Cc0} \\ y_{Cc0} \\ z_{Cc0} \end{pmatrix}. \quad (2.10)$$

After eye rotation the cornea curvature centre can be given by:

$$\mathbf{r}_{\mathbf{Cc}} = \mathbf{A}\mathbf{r}_{\mathbf{Cc0}}. \quad (2.11)$$

Similarly, we can find new coordinates of pupil edge points $(x_I, y_I, z_I)^T$. We assume that the position of image of this point in image sensor is defined by the ray, which propagates from the pupil edge point and after refraction in the cornea-air boundary has the direction, which is parallel to the camera optical axis. If the boundary is a plane then the task to find the ray way is simple. Because the boundary is spherical, the task become complex.

First we will find the point on the cornea surface (x_C, y_C, z_C), where refraction occurs. For this we will select initial ray propagation direction defined by k_{yIris}, k_{zIris}. We can write system of next equations:

$$y_C = y_I + k_{yIris}, \quad (2.12)$$

$$z_C = z_I + k_{zIris}, \quad (2.13)$$

$$(x_C - x_{Cc})^2 + (y_C - y_{Cc})^2 + (z_C - z_{Cc})^2 = R_C^2. \quad (2.14)$$

Further we denote k_{yIris} and k_{zIris} shortly k_y and k_z.

After solving of equations (2.12)-(2.14), we obtain:

$$x_C = \frac{p_2 + \sqrt{p_3}}{p_1}; \qquad (2.15)$$

where

$$p_1 = 1 + k_y^2 + k_z^2; \qquad (2.15)$$

$$p_2 = (k_y^2 + k_z^2)x_I + k_y(y_{rR} - y_I) + k_z(z_{rR} - z_I) + x_{rR}; \quad (2.16)$$

$$\begin{aligned} p_3 = {} & R_C^2 p_1 - (k_y^2 + k_z^2)x_{Cc}^2 - (1 + k_z^2)y_{Cc}^2 - (1 + k_y^2)z_{Cc}^2 + 2k_y x_{Cc} y_{Cc} + 2k_z x_{Cc} z_{Cc} + 2k_y k_z y_{Cc} z_{Cc} + \\ & + 2\left[(k_y^2 + k_z^2)x_{Cc} - k_y y_{Cc} - k_z z_{Cc}\right]x_I + 2\left[-k_y x_{Cc} + (1 + k_z^2)y_{Cc} - k_y k_z z_{Cc}\right]y_I + \\ & + 2\left[-k_z x_{Cc} - k_y k_z y_{Cc} + (1 + k_y^2)z_{Cc}\right]z_I - (k_y^2 + k_z^2)x_I^2 - (1 + k_z^2)y_I^2 - (1 + k_y^2)z_I^2 + \\ & + 2k_y x_I y_I + 2k_z x_I z_I + 2k_y k_z y_I z_I \end{aligned} \qquad (2.17)$$

If we calculate x_C by (2.15), then we can obtain y_C, z_C from Eqn (2.12)-(2.13). Projection of point (x_C, y_C, z_C) to the plane of the image sensor gives us the image of selected pupil edge point. The problem is that we do not know parameters k_y, k_z. We can them find applying the refraction law and optimisation process.

First, we assume that refracted ray goes through point (x, y, z). We select that its abscissa is

$$x = x_C + 1. \qquad (2.18)$$

Then we calculate three vectors $\mathbf{v_1}$, $\mathbf{v_2}$, $\mathbf{v_3}$. Vector $\mathbf{v_1}$ goes from point (x_I, y_I, z_I) to (x_C, y_C, z_C) and represents incident ray. Its components:

$$\mathbf{v_1} = \begin{pmatrix} x_C - x_I \\ y_C - y_I \\ z_C - z_I \end{pmatrix}. \qquad (2.19)$$

Second vector v2 goes from point (x_C, y_C, z_C) to (x, y, z) and represents refracted ray. Its components:

$$\mathbf{v}_2 = \begin{pmatrix} x - x_C \\ y - y_C \\ z - z_C \end{pmatrix}. \qquad (2.20)$$

Third vector $\mathbf{v_3}$ connects cornea curvature centre with point on cornea surface and represents normal to surface. Its coordinates:

$$\mathbf{v}_3 = \begin{pmatrix} x_C - x_{Cc} \\ y_C - y_{Cc} \\ z_C - z_{Cc} \end{pmatrix}. \qquad (2.21)$$

According to refraction law, all three vectors are coplanar. Their components comply equation:

$$\begin{vmatrix} v1x & v1y & v1z \\ v2x & v2y & v2z \\ v3x & v3y & v3z \end{vmatrix} = 0; \qquad (2.22)$$

where | | stands for determinant. Substituting vector components from (2.19), (2.20), (2.21) into (2.22), we obtain:

$$\begin{vmatrix} x_C - x_I & y_C - y_I & z_C - z_I \\ x - x_C & y - y_C & z - z_C \\ x_C - x_{Cc} & y_C - y_{Cc} & z_C - z_{Cc} \end{vmatrix} = 0 . \quad (2.23)$$

Cosine of incident angle α can be found from vectors $\mathbf{v_1}$ and $\mathbf{v_3}$ by equation:

$$\cos\alpha = \frac{\mathbf{v_1}\mathbf{v}_3}{\|\mathbf{v_1}\|\|\mathbf{v}_3\|} = \frac{(x_C - x_I)(x_C - x_{Cc}) + (y_C - y_I)(y_C - y_{Cc}) + (z_C - z_I)(z_C - z_{Cc})}{\sqrt{(x_C - x_I)^2 + (y_C - y_I)^2 + (z_C - z_I)^2}\sqrt{(x_C - x_{Cc})^2 + (y_C - y_{Cc})^2 + (z_C - z_{Cc})^2}} . \quad (2.24)$$

Cosine of refraction angle β can be found from vectors $\mathbf{v_2}$ and $\mathbf{v_3}$ by equation:

$$\cos\beta = \frac{\mathbf{v}_2\mathbf{v}_3}{\|\mathbf{v}_2\|\|\mathbf{v}_3\|} = \frac{(x - x_C)(x_C - x_{Cc}) + (y - y_C)(y_C - y_{Cc}) + (z - z_C)(z_C - z_{Cc})}{\sqrt{(x - x_C)^2 + (y - y_C)^2 + (z - z_C)^2}\sqrt{(x_C - x_{Cc})^2 + (y_C - y_{Cc})^2 + (z_C - z_{Cc})^2}} . \quad (2.25)$$

Snell's law gives equation:

$$n_1 \sin\alpha = n_2 \sin\beta , \quad (2.26)$$

where n_1=1.3375 and n_2=1.

Four equations: (2.23), (2.24), (2.25), and (2.26) allow us to find unknowns α, β, y, z. Then we can calculate refracted ray direction coefficients k_{yr} and k_{zr} :

$$k_{yr} = \frac{y - y_C}{x - x_C} ; \quad (2.27)$$

$$k_{zr} = \frac{z - z_C}{x - x_C} . \quad (2.28)$$

They must be equal to the camera's optical axis direction coefficients k_{yCam}, k_{zCam}.

Optimisation process starts from guess of coefficients k_{yIris}, k_{zIris}, which we need to use in equations (2.12), (2.13). It ends, when these coefficients yield direction of refracted beam expressed by k_{yCam}, k_{zCam}. We obtain criteria for optimization in form:

$$J = \sqrt{(k_{yr} - k_{yCam})^2 + (k_{zr} - k_{zCam})^2} . \quad (2.29)$$

Below are some results of simulation with proposed model. The pupil size changes in the image versus the camera slant angle results are presented in Figure 2.5. Here also is shown cosine function. One could expect change of pupil size by cosine function, if there is no light ray refraction. Presence of refraction on the cornea surface causes faster decrease of the pupil size versus the camera slant angle.

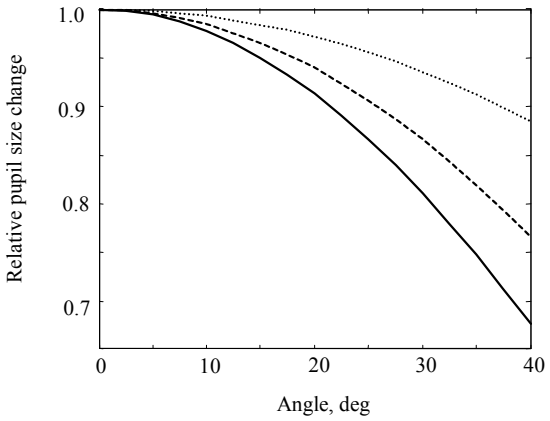


Figure 2.5: Changes of vertical size of pupil versus camera slant angle, when pupil diameter – 5 mm (solid line – model, dash-dotted – cosine function, dotted – additional change to cosine function).

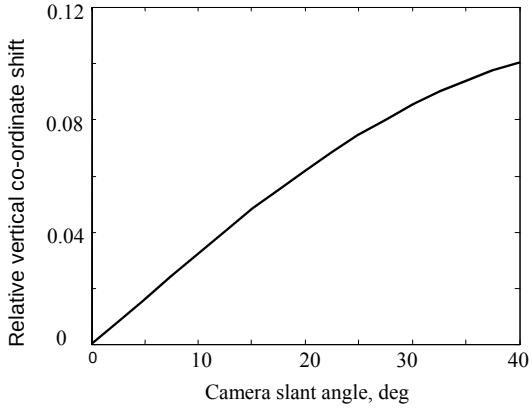


Figure 2.6: Pupil centre vertical coordinate, calculated as pupil mass centre, shift in simulated image from iris centre. Pupil diameter is 5 mm. Changes are normalized to pupil diameter.

The pupil area in the image is non-linearly distorted. The degree of distortion depends on the slant angle. Because of this pupil centre calculated as pupil area mass centre is shifted from the iris centre, which contour can be tracked without refraction. The results are shown in Figure 2.6. Simulation reveals that pupil size changes could decrease the accuracy of gaze tracking.

2.4 Corneal Reflection

The front surface of the cornea (over its central 25 degrees) can be approximated by a spherical section. Reflections of a bright object from this surface form a virtual image behind the surface which can be imaged and photographed. The position of the corneal reflections, commonly seen as the highlights in the eye, is a function of eye position.

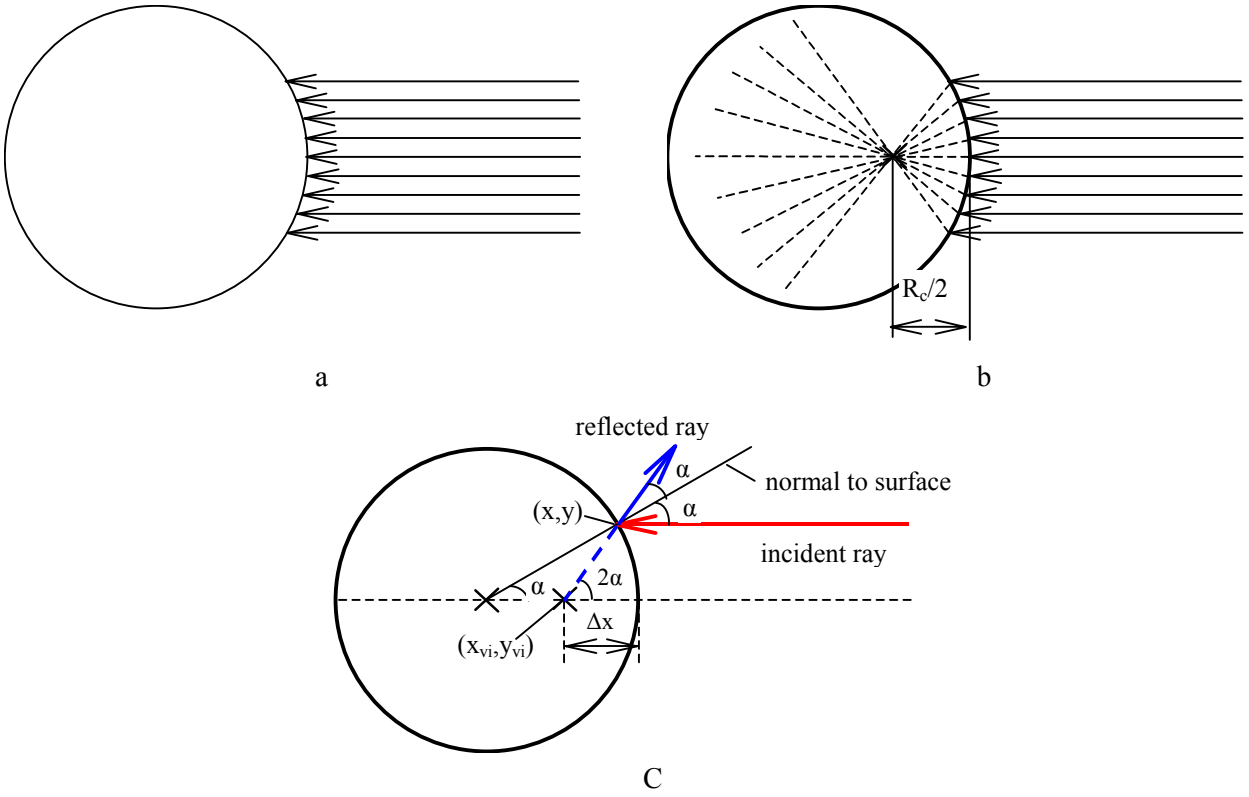


Figure 2.7 : Formation of virtual image of corneal reflection.
(a) incident parallel rays, (b) elongations of reflected rays, (c) calculation of virtual source position.

Now we calculate the position of the virtual image. Assume that a beam of parallel rays incident on the cornea surface (see Figure 2.7a). The continuations of reflected beams to opposite direction intersect in one point (see Figure 2.7b). It is virtual image of source. The symmetry of the formation reveals that the virtual image is on line, which runs through the cornea curvature centre. We will find its position from the frontal surface. The incident angle α depends on the vertical coordinate of the point on the cornea surface:

$$\sin\alpha = y / R_c . \quad (2.30)$$

Coordinate x is expressed as:

$$x = R_c \cos\alpha . \quad (2.31)$$

Because the continuation of the reflected ray intersects with central line by angle 2α, the virtual image is shifted to the centre of cornea curvature by Δx:

$$\Delta x = \frac{y}{tg2\alpha} . \quad (2.32)$$

From (2.30)

$$y = R_c \sin\alpha . \quad (2.33)$$

Substitution into (2.32) yields:

$$\Delta x = \frac{R_c \sin\alpha}{tg2\alpha} . \quad (2.34)$$

If α is small, then:

$$\begin{cases} \sin\alpha \approx \alpha; \\ \cos\alpha \approx 1; \\ tg2\alpha \approx 2\alpha. \end{cases} \quad (2.35)$$

Substituting (2.35) into (2.34) we obtain:

$$\Delta x \approx \frac{R_c}{2} \qquad (2.36)$$

We obtained that the virtual image of the corneal reflection is in the middle between the cornea curvature centre and the closest point to the light source on the surface on cornea. Further, we will find coordinates of the virtual image, when the coordinates (x_{si}, y_{si}) of i-th light source are known. Then the direction coefficients (k_{yvi}, k_{zvi}) of the line connecting the cornea curvature centre with the light source are expressed as:

$$k_{yvi} = \frac{y_{si} - y_{Cc}}{x_{si} - x_{Cc}}, \qquad (2.37)$$

$$k_{zvi} = \frac{z_{si} - z_{Cc}}{x_{si} - x_{Cc}}. \qquad (2.38)$$

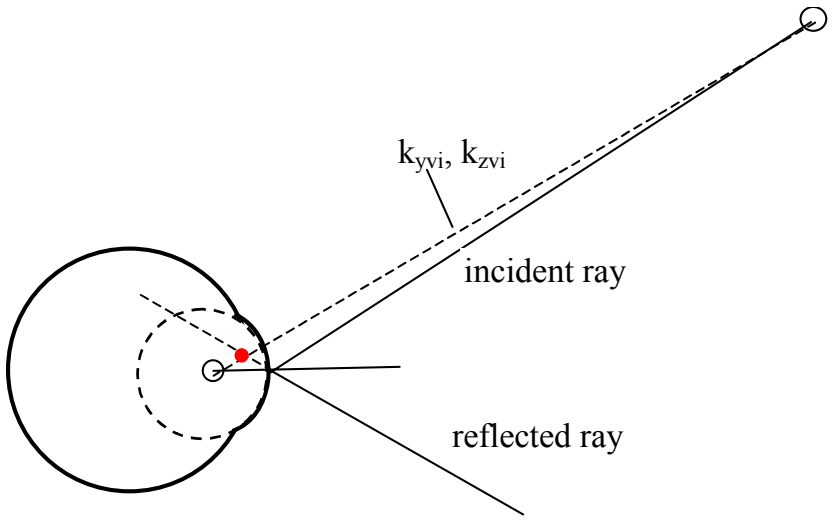


Figure 2.8: Finding of virtual image, when the coordinates of light source are known.

Without reduction of generality, we can select the beginning of coordinates system in centre of cornea curvature. It is: x_{Cc}=0, y_{Cc}=0, z_{Cc}=0. We denote the coordinates of the virtual image as (x_{vi},y_{vi}, z_{vi}). Because the distance of virtual image of the cornea curvature centre is R_C/2:

$$x_{vi}^2 + y_{vi}^2 + z_{vi}^2 = \left(\frac{R_c}{2}\right)^2 . \qquad (2.39)$$

Coordinates yvi and zvi can be expressed using k_{yvi}, k_{zvi}:

$$y_{vi} = k_{yvi} x_{vi}; \quad (2.40)$$

$$z_{vi} = k_{zvi} x_{vi}. \quad (2.41)$$

From system of equations we find x_{vi}:

$$x_{vi} = \frac{R_c}{2\sqrt{1 + k_{yvi}^2 + k_{zvi}^2}}. \quad (2.42)$$

Then:

$$y_{vi} = k_{yvi} x_{vi}; \quad (2.43)$$

$$z_{vi} = k_{zvi} x_{vi}. \quad (2.44)$$

The projection of virtual sources into camera's plane gives locations of the corneal reflections in the image. Most important question is how some corneal reflections can help to estimate the user distance from the monitor's screen. It seems that the distance between two corneal reflections in the image can help (see Figure 2.9).

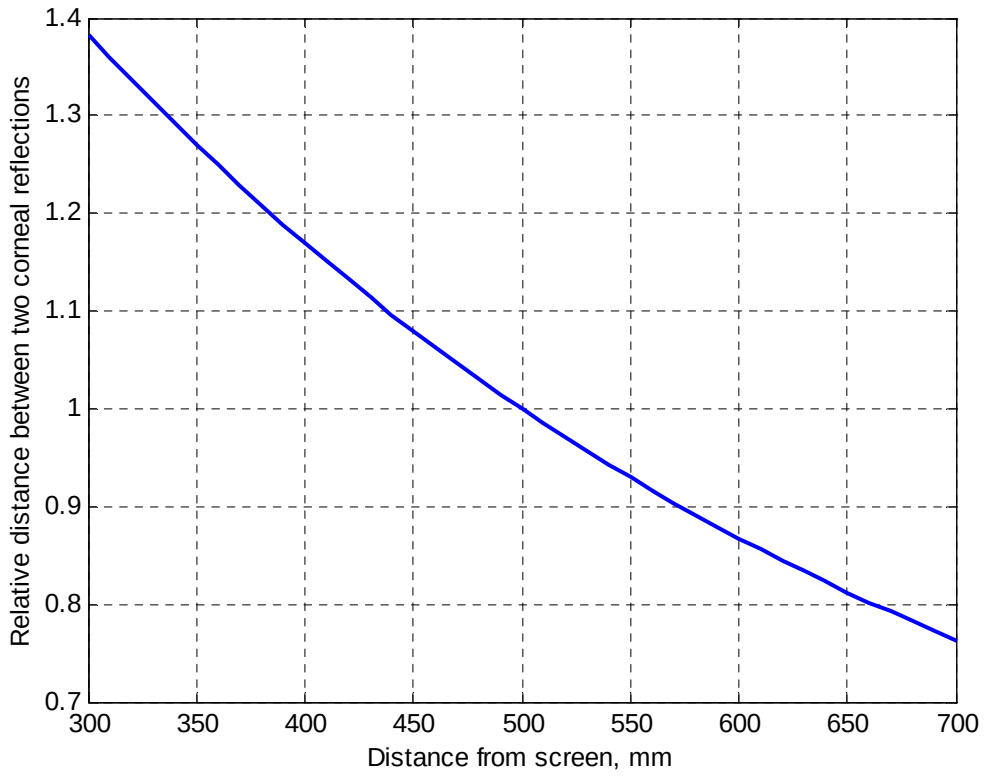


Figure 2.9: Relative distance between two corneal reflection versus eye distance from screen. Y axis normalized to distance from screen L=500 mm.

3 Algorithms for IR Eye Tracking

3.1 The "Starburst" Eye Tracking Algorithm

The "Starburst" algorithm [17] is an algorithm for detecting and measuring the position of the pupil and corneal reflex (CR) in an image of the eye region. It is used in the open-hardware, open-source "openEyes" eye tracker [18, 19]. Starburst is a hybrid algorithm that uses both feature-based and model-based approaches. A good estimate for the position of the pupil contour is found by looking for points on the contour of the pupil, then fitting an ellipse to these; the position of this ellipse is then fine-tuned using a model-based technique that maximizes the ratio between pixel intensities on the outside and the inside of the ellipse. A similar mix of model-based and feature-based techniques is used to measure the position of the CR.

3.1.1 Corneal reflex detection and removal

The first step in the Starburst algorithm is to find the corneal reflex (CR) in the image. The CR is then removed from the image to ease the task of subsequent processing stages. For example, CR removal prevents erroneous detections of pupil contour points on the contour of the CR.

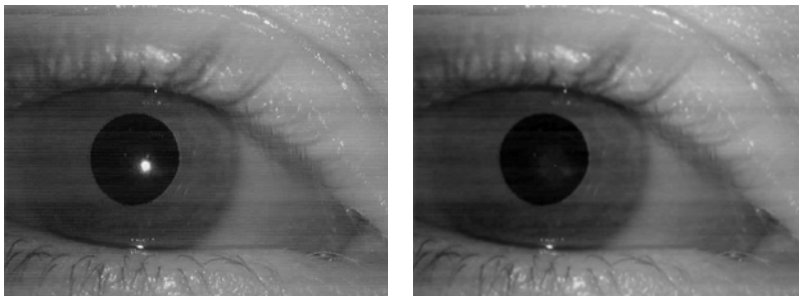


Figure 3.1: (left) Sample image of eye region with corneal reflex (CR); (right) Eye region with CR removed. (Courtesy of D. Parkhurst.)

The CR is found using an adaptive thresholding technique that starts with a high threshold to find the corneal reflex, which is assumed to be the largest connected region with pixel values greater than the threshold. It is unlikely, however, that this region will contain all of the pixels in the CR, because the brightness of the CR decreases towards its border. Therefore, the threshold is now decreased successively until the ratio between the pixel count of the largest above-threshold region and the pixel count of the other regions becomes maximal. The rationale behind this is that the CR region will initially grow rapidly as the threshold is lowered; at some point, when the segmented region reaches the border of the CR, this region will stop growing, while the false positives continue growing. Therefore, the threshold should not be decreased beyond this point.

To improve the accuracy of the CR position measurement, the algorithm fits a circle to the CR by maximizing an edge measure integrated over the contour of the circle. The CR is then removed from the image by filling the segmented CR region with values interpolated from the border of the region (see Figure 3.1).

3.1.2 Pupil contour detection

The pupil contour detection step starts with an initial guess for the pupil centre. This can be obtained from the position of the pupil in the previous frame; for the first frame, the position of the pupil can either be initialized manually, or the centre of the eye region can simply be used as an initial pupil centre guess.

The algorithm shoots rays out radially from the initial pupil centre guess and finds contour points on these rays where the derivative exceeds a certain threshold. Because the algorithm is designed for dark-pupil images, it requires the derivative to be positive for a contour point to be detected. From each of the detected contour points, the algorithm shoots a fan of secondary rays back into the pupil to find additional contour points (see Figure 3.2).

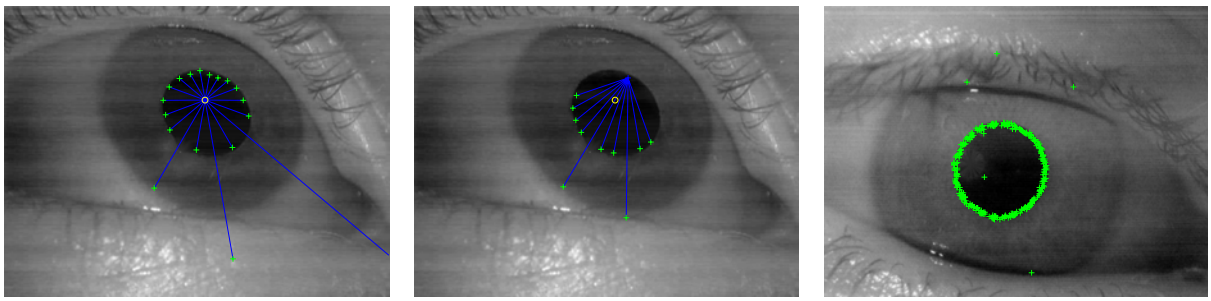


Figure 3.2: (left) Rays shot outwards from the initial pupil centre guess and detected contour points; (centre) Secondary rays shot back from contour point with newly detected contour points; (right) contour points detected in the two phases. (Courtesy of D. Parkhurst.)

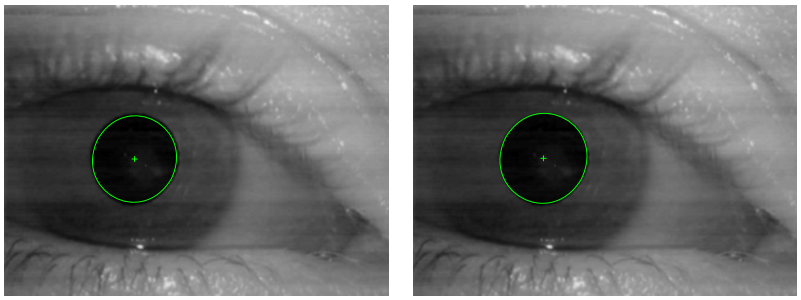


Figure 3.3: (left) Result of RANSAC ellipse fitting; (right) ellipse found by maximizing edge strength measure. (Courtesy of D. Parkhurst.)

An ellipse is then fit to the detected contour points using a RANSAC (Random Sample Consensus) algorithm (see Figure 3.3). The idea behind the RANSAC algorithm is to take a certain number of random samples of five points each from the set of contour points, fit an ellipse through these points (five is the minimum number of points needed to determine the ellipse uniquely), then determine which of the other contour points lies on the ellipse (with a certain tolerance); this set of points is called the consensus set. The algorithm then chooses the ellipse with the largest consensus set and computes the final result as a least-squares ellipse fit to this whole consensus set. The advantage of using the RANSAC algorithm as opposed to fitting an ellipse to the set of all contour points is that it is much more robust in the presence of outliers.

In a final step, the position of the ellipse obtained through the RANSAC algorithm is fine-tuned in a model-based approach by maximizing a measure of edge strength, integrated over the contour of the ellipse (see Figure 3.3).

3.1.3 Gaze estimation through homographic mapping

To calculate the user's gaze position from the measured position of the pupil and the corneal reflex, the algorithm uses a linear homographic mapping. This mapping, represented as a matrix *H*, takes the difference vector between the pupil centre and the CR and maps it to the gaze position in the scene. Homogeneous coordinates are used to represent the difference vector and the gaze position so that the mapping can perform shifts by a constant offset.

To determine the entries of the matrix *H*, the user is asked to fixate several points with known positions in a calibration phase. This results in a number of known correspondences between pupil-CR difference vectors and gaze positions that can be used to solve for *H*. Since *H* has eight degrees of freedom, a minimum of four correspondences are needed, but more are typically used; in the latter case, the algorithm determines the mapping *H* that minimizes the error on the calibration points.

3.2 Coordinates Averaging Method[3]

3.2.1 Algorithm

The detection of pupil center in the image of eye is the most important step for video-based eye tracking method, because full error directly depends on pupil center coordinates errors. The pupil is the largest dark area in the image of eye and it can be distinguished from the surrounding iris by brightness threshold value. If pixel brightness value is less than threshold value, then this pixel is assigned to pupil. If the coordinates of pupil center will be evaluated in integer values of pixels, this cause loss of some information, and method errors will be large. Because we want achieve good accuracy, pupil edge must be located with subpixel accuracy. In the region between iris and pupil brightness from pixel to pixel changes monotonically. It helps us to evaluate edge points with subpixel accuracy. In the transit region pixel brightness is fitted to polynomial function versus coordinate. Then pupil edge point coordinates can be evaluated precisely. It is evident that the errors are inverse proportional to brightness gradient.

The pupil center coordinates can be determined from pupil edge points position by some different methods. We used two distinct methods. One of them is widely used computer vision method, it is points fitting by circle method. Shorter we call it as circle approximation method. Another method is novel. It is based on averaging of coordinates of pupil contour. Further we named it as coordinates averaging method.

The complete pupil edge is defined after two steps: (1) horizontal scanning and (2) vertical scanning. Although scanning in both directions allows to get better performance, it is enough to scan in one direction for circle approximation method. In opposite, the second method needs of both scannings

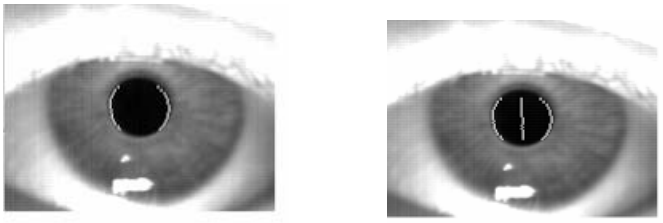
The image processing stages results are shown in Figure 3.4. After first stage pupil edge points were detected in scanning lines (Figure 3.4a). For lines with pupil the two points were defined. At the second stage the average of edge points coordinates in each scanning line was calculated (Figure 3.4b). A set of new points can be approximated by a vertical line. The result of approximation is equation coefficients v_0 and v_1 of vertical line:

$$y = v_0 + v_1 x; \tag{3.1}$$

which are obtained by points fitting to line exploiting least squares method.

[3] Section 3.2 is based on a draft manuscript; the final paper was published as:

Daunys, G. and Ramanauskas, N. (2004) The accuracy of eye tracking using image processing. *Proceedings of the Third Nordic Conference on Human-Computer interaction, NordiCHI '04,* vol. 82. ACM Press, New York, NY, 377-380. DOI= http://doi.acm.org/10.1145/1028014.1028074



a) b)

Figure 3.4. Two stages in pupil image processing: a) after pupil edge points detection, b) after pupil vertical axis detection.

Second step is analogous to the first, but now scanning in vertical direction is processed. Fitting to line gives us a equation of horizontal line:

$$y = h_0 + h_1 x \ . \tag{3.2}$$

To find the center of the pupil we solve an equation system consisting of equations (13) and (14). We obtain:

$$x_0 = \frac{h_0 - v_0}{v_1 - h_1}; \tag{3.3}$$

$$y_0 = v_0 + v_1 x_0 . \tag{3.4}$$

For detection of eye ball angular position coordinates one must do calibration. When the changes of coordinates are small the calibration is linear process. To achieve more general results we shall present results without calibration.

Both described pupil center detection methods were implemented in software using C++ programming language.

3.2.2 Generation of synthetic image sequences

In order to examine the accuracy of pupil detection methods the synthetic image sequences were generated. They were used as input for images analysis programme. During generation of video files the signal formation in the video sensors was simulated. It was taken into account that the voltage of each pixel is proportional to the average intensity of light striking on the active pixel area (we used value 64 percents of full pixel area). Also the video signal was discretized into 256 voltage levels in order to examine quantisation errors during analog-to-digital conversion.

Every file contained forty frames. We used three different trajectories for imaginary eye movement:

- the eye had moved straight down for 0.05 of pixel per frame;
- the eye had moved straight right for 0.05 of pixel per frame;
- the eye had moved by diagonal line to left-up direction by 0.05 of pixel in horizontal and vertical directions;

The attempt to evaluate the influence of different artifacts in video images was made. Usually odd and even lines of image have a different brightness despite the progressive scan method. So we produced video files with such discrepancy. Sequences of images when pupil is partially closed with eyelid were generated also. For different conditions images with superimposed white noise with various standard deviations (σ) were created.

In the images without artifacts the full change of brightness between pupil and surrounding iris was 64 units.

3.2.3 Results

The produced files with synthetic eye images were processed by video image analysis software with implemented both pupil detection methods. The errors as difference between detected pupil center and actual one were calculated.

At first, we examined influence of pupil brightness threshold on detection errors. We equated the difference between iris brightness and pupil brightness to 100%. The analysis of noiseless images revealed that the threshold value in the range from 20% to 70% has no influence on the standard deviation on error. But the threshold becomes important when white noise with big standard deviation is superimposed on image.

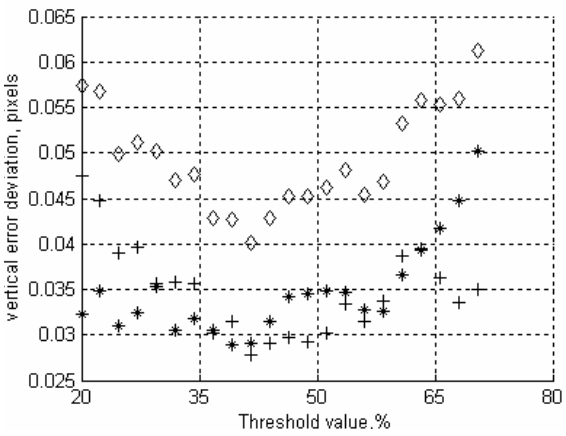


Figure 3.5. Standard deviations of error versus threshold value ('+' – horizontal error; '*' –vertical error; '◊' – full error).

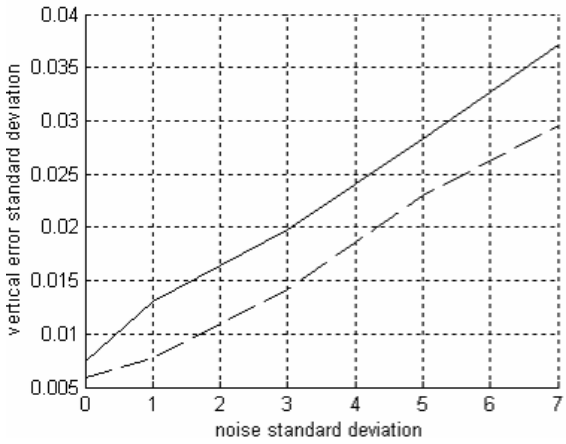


Figure 3.6. Standard deviation of vertical error versus standard deviation of noise.
'--' – coordinates averaging method; '—' – circle approximation method.

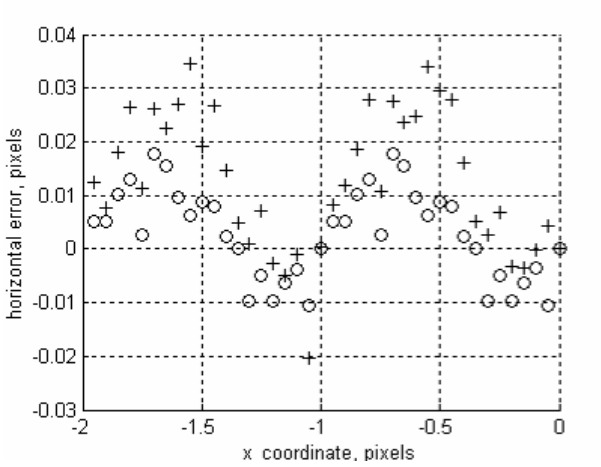


Figure 3.7. Horizontal error versus actual horizontal coordinate of pupil center x_0
('+' – circle approximation method; 'o' – coordinates averaging method).

In Figure 3.5 the error and its horizontal and vertical components standard deviations were plotted versus threshold value, when white noise σ=5. We can see from the plot that the minimum error is reached when the threshold value is about 40%. For further analysis we used optimal threshold value.

Figure 3.6 shows standard deviation of vertical error relative to white noise standard deviation. The different pupil center detection methods are represented by different lines. The slope of both lines is very close.

The results of analysis of movement by diagonal line are shown in Figure 3.7 and Figure 3.8. Here errors of pupil center coordinates are plotted versus actual pupil coordinates

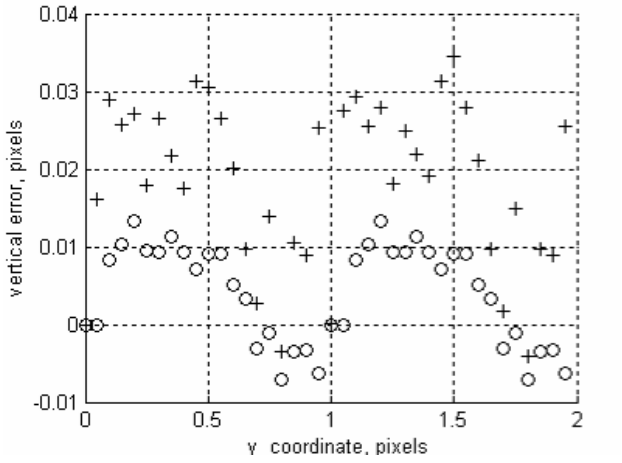


Figure 3.8. Vertical error versus actual vertical coordinate of pupil center y_0
('+' – circle approximation method; 'o' – coordinates averaging method).

3.2.4 Discussion

Analysis of results shows that both pupil center detection methods give similar errors when eye position is straight a head. They can be used for pupil center coordinates detection in software for videooculography. Both methods have their advantages and disadvantages.

Advantages of coordinates averaging method are dominating, when full pupil is open or pupil edge is damaged by small diameter artifacts. This method is faster.

Also it is important that the errors are more predictive than in case of circle approximation method.

The accuracy of videooculography method was investigated. The errors of measurements in low noise conditions can be less than 1 arc minute. The proposed novel pupil center detection method has the same accuracy as approximation by circle method, but has better computational performance.

3.3 Algorithm for Pupil-Glint Vector Detection in a Bright Pupil Eyetracking System[4]

This kind of algorithm employs infrared lighting coaxially disposed respect to the optical axis of the camera that is fitted under the screen see Figure 3.9.

The obtained images, Figure 3.9, have two special characteristics. The pupil looks brighter than most of the rest of the image and secondly it assures the existence of a bright point corresponding to the glint of the diode in the cornea. The position of both points is directly related with the gaze position. This technique known as bright pupil presents two concrete problems. On one hand, in the situation when there is a lot of external lighting in the scene, the pupil looses its contrast respect the rest of the image, what makes more difficult to determine its boundaries. On the other hand it appears multiple glints when the users use glasses or contact lenses what prevent the algorithm from determining which of them is the corneal reflection.

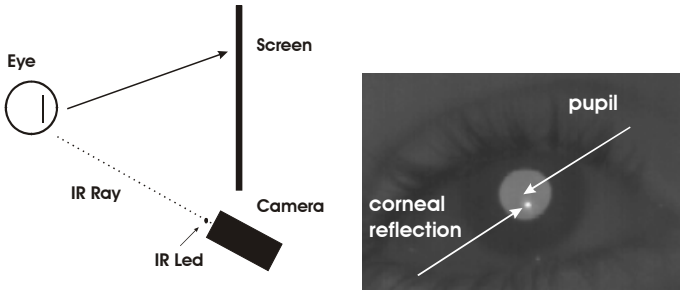


Figure 3.9: Diagram of the infrared Video-Oculography system and an image obtained with the system.

3.3.1 Algorithm description

The aim of the algorithm is to determine both important characteristics using the information recorded by the image acquisition system: that is to say, the centre of the pupil and the corneal glint. This detection must be as robust as possible respect to external lighting changes as well as to other spurious reflections.

The algorithm covers two phases: a preliminary search of both interesting points in the eyes and a subsequent refinement in the calculation of the centre of the pupil and the glint. The first phase leads to a suitable fit of the processing window size, obtaining a notable reduction in calculation time. Let's describe in detail and separately each of these phases.

Preliminary Detection.

Observing the images taken by the camera, it can be checked that whether the pupil as the glint has certain characteristics that make them be different from the rest of the objects in the image.

[4] Section 3.3 is based on a draft manuscript; the final paper was published as:

Goni, S., Echeto, J., Villanueva, A., and Cabeza, R. (2004). Robust Algorithm for Pupil-Glint Vector Detection in a Video-oculography Eyetracking System. *Proceedings of the 17th international Conference on Pattern Recognition, (ICPR'04)*, Volume 04. IEEE Computer Society, Washington, DC, 941-944.

DOI= http://dx.doi.org/10.1109/ICPR.2004.776

- Pupil
  - Average bright level higher than background's and lower than glint's.
  - Average area: bigger than glint's and smaller than background's.
  - High compactness
- Glint
  - High bright level
  - Small area
  - Close to the pupil

Taking into account these properties it's time to get the estimated location of the pupil and the glint. The algorithm employs at different steps certain configuration values that have been adjusted and experimentally validated for the used system and for a standard working session. Anyway these can be easily fitted whether any of the characteristics of the tracking changes.

As a previous step to this detection, the image is binarized with two pre-established thresholds, wide enough as to cover all the possible situations of illumination in which the system could be used. These values are not critic due to, as we'll see later, the system has the capacity to vary them dynamically as they are evolved. They are, therefore, two initial values that allow to assure that the limits of the pupil are within such segment and whose default values have been experimentally validated. On the other hand and in the worst case, they could be adjusted at the start of each session.

Once the image is binarized, an open and close operation is carried out with the purpose of eliminating the noise and the possible artefacts present in the image. As result of the two previous steps a group of blobs is obtained, among which it'll be necessary to distinguish the pupil. As first criterion of selection a filter by area is made. The result will be all those blobs whose sizes were within the two pre-established values (between 600-4000 pixels). The last step is a selection by compactness. Due to its circular shape, the pupil presents a high compactness respect the rest of blobs present in the image. Calculating the compactness index of the remaining blobs, the one corresponding to the pupil is finally obtained (see Figure 3.10). From this blob its mass centre as well as the contact points window, that is the minimum rectangular window containing the pupil, are extracted.

To minimize the calculation time of this first step in the algorithm (that will be proportional to the image area to be processed), the information of the former image is used. More precisely the position of the glint is used, so that a window of 180x135 pixels is centred at the position of the glint of the mentioned image, based on the hypothesis that the pupil can not vary its positions too much from image to image. It has been demonstrated that this size assures that the pupil is found in the searching area allowing a low computation time. Obviously, for the initial image of the session the detection will be made in the whole image acquired by the camera.

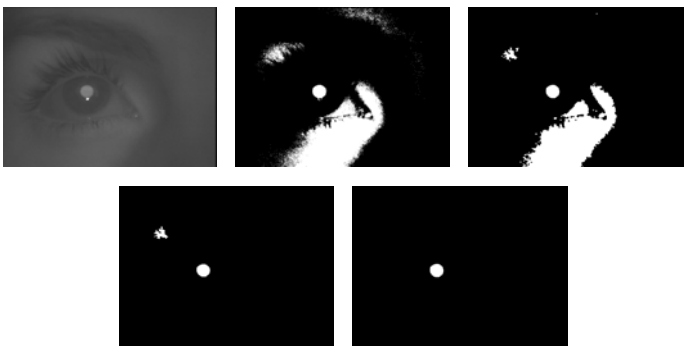


Figure 3.10: Pupil detection process. Thresholding, size filtering and compactness selection.

Once the estimated position of the pupil is carried out, the glint is searched. For this purpose, first of all the image is binarized with the highest threshold employed in the previous step. The obtained blobs are filtered by area eliminating the elements with an area lower than 5 pixels and higher than 200 pixels. At last, to be able to discern among multiple reflections, the closest to the mass centre of the recently calculated pupil is chosen (see Figure 3.11). Once the candidate has been obtained, its mass centre is estimated as well as its contact points.

With the purpose to minimize the processing time, an increment of 40 pixels is added to the contact points window calculated for the pupil to perform the glint searching. It has been widely demonstrated that the glint is always found in such area.

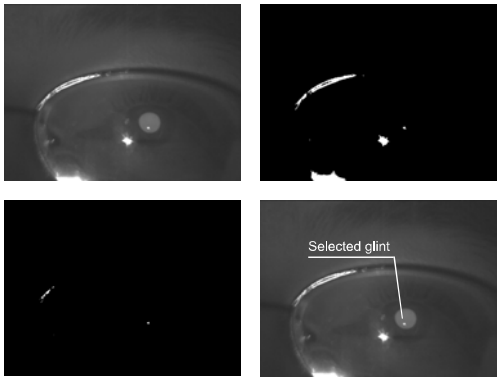


Figure 3.11: Glint detection process. Thresholding, size filtering and distance to the previously located pupil.

Calculation of thresholds

In order to perform the exact calculation of the two pursued image characteristics, an exhaustive calculation of the threshold values is carried out.

A window is located containing not only the pupil but also the glint to obtain later a histogram from itself as most defined as possible. The contact points of each calculated blob at former steps are used to construct the window leaving a safety margin of 20 pixels to put away possible approximation errors in previous steps. In this way it's assured that the pupil and the glint are within the window.

The histogram of this window is extracted obtaining the thresholds that limit the grey levels of the pupil and the glint. In order to locate in this histogram the maximums corresponding to the background and the pupil the threshold for the pupil obtained in the former image is employed. It divides the grey level distribution in two areas clearly different: the right area corresponds with the pupil and the left area with the background and their maximums are identified as pupils and backgrounds peaks respectively.

The threshold of the pupil is to be determined at some value between the maximums previously located. However this determination could result awkward because of the ringing existing between those maximums. In order to palliate as far as possible this limitation, the ringing or intermediate area is estimated and the average level of such uncertainty area is fixed as the threshold of the pupil. The ringing interval is defined as the interval delimited by the grey levels whose values in the histogram are below a certain value. This limit is calculated as the arithmetic average between two histogram values. The first one will correspond to the lowest maximum calculated, i.e. pupils or backgrounds peak. The other value will be the minimum reached by the histogram within the interval between those maximums. As it's been already commented, the level of the pupil is chosen as the middle point of the ringing interval (see Figure 3.12).

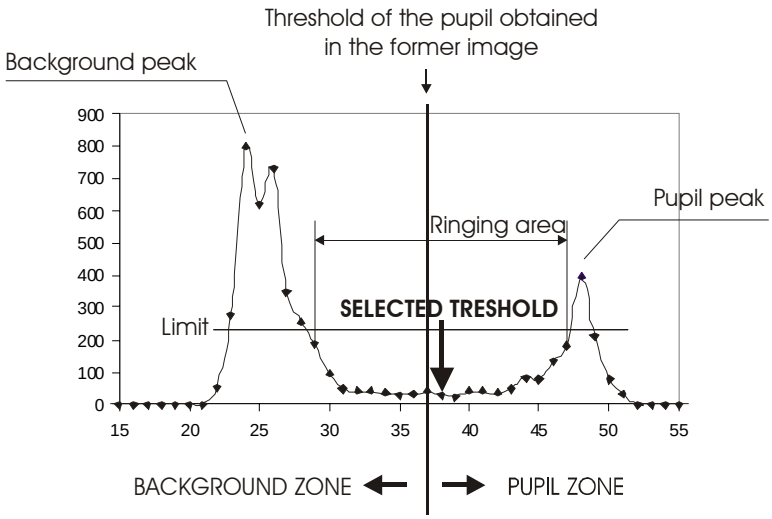


Figure 3.12: Typical histogram with its associated parameters.

Once obtained the threshold of the pupil, the one for the glint is to be got. Whereas the background and the pupil present a very clear peak in the histogram, the glint doesn't present any characteristic that makes it different because of its low number of pixels. Its only characteristic reflected in the histogram is that its level is higher than pupil's. In this case, the criterion is simple: it is chosen the grey level with next property: the threshold and the next two levels are, at most, represented each one by two pixels in the image. The first level of those three will be chosen as the threshold of the glint. This method has shown to be effective for glint thresholding with the working conditions of the system.

Precisely determination of the centres

Once the thresholds of the pupil and the glint have been extracted, the centre of both objects is going to be obtained in a very precisely way. Each of these calculations will be restricted by a window containing only the object in question. In this way the processing time is reduced and the precision is increased.

For the calculation of the centre of the glint, it was employed the window obtained in the preliminary detection enlarged by 10 pixels. The image with the threshold calculated from the former section is binarized and the mass of the resulting blob centre is calculated. This one will be the definitive centre of the glint. Further coordinates averaging algorithm, which was given in Section 3.2, is used.

Once the centres of the pupil and glint are obtained, it's possible to determine the direction of the user gaze, whether the whole system has been previously calibrated. It's important to point out the fact that the information obtained with success from the processing of an image, threshold and centres, is used as initial parameters for the following processing tasks, based on the hypothesis that the parameters involved in the process don't vary too much from one image to other.

4 Tracking in Visual Light

4.1 Introduction[5]

Eye tracking and detection methods fall broadly within three categories, namely deformable templates, appearance-based, and feature-based methods [20, 21]. Deformable template and appearance-based methods rely on building models directly on the appearance of the eye region while the feature-based methods rely on extraction of local features of the region. The latter methods are largely bottom up while template and appearance-based are generally top-down approaches. That is, feature based methods rely on fitting the image features to the model while appearance and deformable template-based methods strive to fit the model to the image.

In general appearance models detect and track eyes based on the photometry of the eye region. A simple way of tracking eyes is through template-based correlation. Tracking is performed by correlation maximization of the target model in a search region. The appearance of eye regions share commonalities across race, illumination and viewing angle.

Deformable template-based method [22, 23, 24] rely on a generic template which is matched to the image. In particular deformable templates [22], construct an eye model in which the eye is located through energy minimization. In the experiments it is found that the initial position of the template is critical. Another problem lies in describing the templates. Whenever analytical approximations are made to the image, the system has to be robust to variations of the template and the actual image.

The deformable template-based methods seem logical and are generally accurate. They are also computationally demanding, require high contrast images and usually needs to be initialized close to the eye. While the shape and boundaries of the eye are important to model so is the texture within the regions. For example the sclera is usually white while the region of the iris is darker. Hansen et al. [25] propose a method which uses Active Appearance Models for local optimization and a mean shift color tracker for handling larger movements. The Active Appearance Models effectively combines pure template-based methods with appearance methods. While the Active Appearance Model shares some of the problems with template-based methods, these models should in theory be able to handle changes in light due to its statistical nature. In practice they are quite sensitive to these changes and especially light coming from the side can have a significant influence on their convergence.

Feature-based methods extract particular features such as skin-color, color distribution of the eye region. Kawato and Tetsutani [26] and Yang et al. [27] use a circle frequency filter and background subtraction to track the in-between eyes area and then recursively binarize a search area to locate the eyes. Herpers et al. [28] utilize Gabor filters to locate and track the features of eyes. They construct a model-based approach which controls steerable Gabor filters: The method initially locates a particular edge (i.e., left corner of the iris) then use steerable Gabor filters to track the edge of the iris or the corners of the eyes. Nixon [29] demonstrates the effectiveness of the Hough transform modelled for circles for extracting iris measurements, while the eye boundaries are modelled using an exponential function. Young et al. [30] show that using a head mounted camera and after some calibration, an ellipse model of the iris has only two degrees of freedom (corresponding to pan and tilt). They use this to build a Hough transform and active contour method for iris tracking using head mounted cameras. Loy and Zelinsky [31] proposes the Fast Radial Symmetry Transform

[5] Section 3.1 is based on (a part of) a paper published as:

Hansen, D. W. and Pece, A. E. (2005). Eye tracking in the wild. *Computer Vision and Image Understanding 98*, 155-181. DOI= http://dx.doi.org/10.1016/j.cviu.2004.07.013

for detecting eyes in which they exploit the symmetrical properties of the face. Hybrid methods that uses statistical learning of the appearance and local features through Haar wavelets have recently been proposed in the framework of boosting [32]. This approach is discussed for eye detection elsewhere in this issue.

Eye tracking methods committed to using explicit feature detection (such as edges) rely on thresholds. Defining thresholds can in general be difficult since light conditions and image focus change. Therefore, methods relying on explicit feature detection may be vulnerable to these changes.

4.2 Deformable Template Matching[6]

Modeling the iris as a circle is well-motivated when the camera pose coincides with the optical axis of the eye. When the gaze is off the optical axis, the circular iris is rotated in 3D space, and appears as an ellipse in the image plane. Thus, the shape of the contour changes as a function of the gaze direction and the camera pose. The objective is then to fit an ellipse to the pupil contour, which is characterized by a darker color compared to the iris. The ellipse is parameterized,

$$\mathbf{x} = (c_x, c_y, \lambda_1, \lambda_2, \theta), \qquad (4.1)$$

where (c_x, c_y) is the ellipse centroid λ_1 and λ_2 are the lengths of the major and minor axis respectively. θ is the orientation of the ellipse.

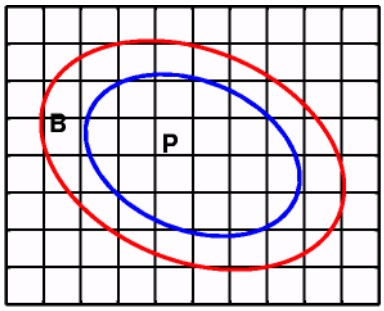


Figure 4.1: The deformable template model. Region *P* is the inner circle, and region *B* is the ring around it.

The model proposed here is based on the relationship between pixel values in two regions, see Figure 4.1. The pupil region *P* is the part of the image *I* spanned by the ellipse parameterized by **x**. The background region *B* is defined as the pixels inside an ellipse, surrounding but not included in *P*, as seen in Figure 4.1. When region *P* contains the entire object, *B* must be outside the object, and thus the difference in average pixel intensity is maximal. To ensure equal weighting of the two regions, they have the same area. The area of the inner ellipse *P* is $A_P = \pi\lambda_1\lambda_2$. The shape parameters of *B* should satisfy the constraint on the area $A_{B/P}$-A_P=A_P. As a consequence, the parameters is defined as $x_B = (c_x, c_y, \sqrt{2}\lambda_1, \sqrt{2}\lambda_2, \theta)$, while x_P is defined as (4.1).

The pupil contour can now be estimated by minimizing the cost function,

$$\varepsilon = \mathrm{Av(P)} - \mathrm{Av(B)}, \qquad (4.2)$$

where Av(P) and Av(P) are the average pixel intensities of the background - in this case the iris - and pupil region respectively. The model is deformed by Newton optimization given an appropriate starting point. Due

[6] Sections 4.2.–4.3 are based on a presentation given in the following national conference:

Martin Vester-Christensen, Denis Leimberg, Bjarne Kjær Ersbøll, Lars Kai Hansen (2005) Deformable Models for Eye Tracking. Presented in *Den 14. Danske Konference i Mønstergenkendelse og Billedanalyse*. Available online at http://www2.imm.dtu.dk/pubdb/views/edoc_download.php/3900/pdf/imm3900.pdf

to rapid eye movements, the algorithm may break down if one uses the previous state as initial guess of the current state, since the starting point may be too far from the true state. As a consequence, we use a simple 'double threshold' estimate of the pupil region as starting point.

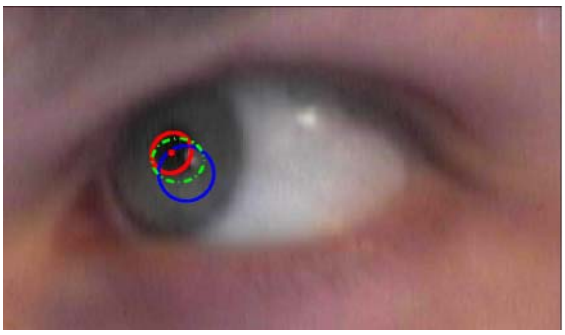


Figure 4.2: The blue ellipse indicates the starting point of the pupil contour. The template is iteratively deformed by an optimizer; one of the iterations is depicted in green. The red ellipse indicates the resulting estimate of the contour.

An example of the optimization of the deformable model is seen in Figure 4.2.

4.2.1 Constraining the deformation

Although a deformable template model is capable of catching changes in the pupil shape, there are also some major drawbacks. Corneal reflections may confuse the algorithm and cause it to deform unnaturally. In the worst case, the shape may grow or shrink until the algorithm collapses.

We propose to constrain the deformation of the model in the optimization step by adding a regularization term. Assuming the parameters defining an ellipse are normally distributed with mean μ, and covariance $\mathbf{\Sigma}$. The prior distribution of these parameters is then defined,

$$p(\mathbf{x}) = N(\mu, \mathbf{\Sigma}) \propto \exp(-\frac{1}{2}(\mathrm{x} - \mu)^T \mathbf{\Sigma}^{-1}(\mathrm{x} - \mu)), \qquad (4.3)$$

where the normalization factor has been omitted. The mean and covariance are estimated in a training sequence. At last the optimization of the deformable template matching method is constrained by adding a regularization term,

$$\varepsilon = \mathrm{Av(P)} - \mathrm{Av(B)} + K(1 - p(\mathrm{x})), \qquad (4.4)$$

where K is the gain of the regularization term.

The relevance of constraining the deformation is visualized in Figure 4.3. A suitable starting point $\mathbf{x}$ is chosen. The pose and orientation are kept fixed, while the shape parameters are varied. In this case the true shape parameters λ_1 and λ_2 are approximately eight. The image confidence as a function of the shape parameters is depicted to the left, while the prior distribution is seen in the middle of Figure 4.3. Combining the image confidence with a prior according to (4.4) yields the constrained estimate, which is depicted to the right in Figure 4.3.

By use of the shape constraints, we incorporate prior knowledge to the solution. The robustness is increased considerably and the parameters are constrained to avoid the algorithm to break down due to infinite increase or decrease of parameters.

The deformable template matching method is seen applied with and without constraints in Figure 4.4. The constrained estimate is seen to be less sensitive to noise due to reflections.

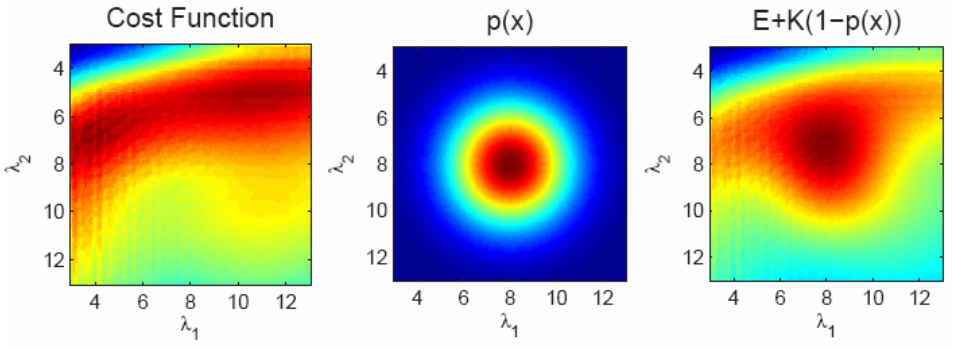


Figure 4.3: Given an appropriate starting point x. The pose and orientation are kept fixed, while the shape parameters are varied. Note that the surface plots are not - as expected - smooth. This is due to rounding in the interpolation when evaluating the image evidence of the deformable template. *(Left)* The image confidence given the state - warmer colours means more likely. *(Middle)* The prior probability is a normal distribution with a given mean value μ, and covariance $\mathbf{\Sigma}$. *(Right)* Combining the image evidence and prior according to (4) yields the constrained estimate.

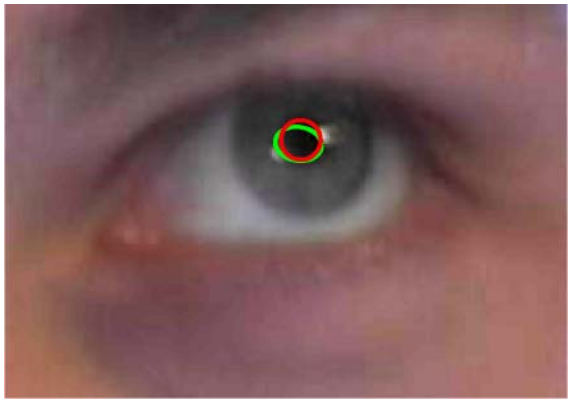


Figure 4.4: The deformable template matching method applied without constraints is seen in green, while the red ellipse depicts the constrained version. The constrained estimate is seen to be less sensitive to noise due to reflections.

4.3 EM Contour Tracking

The iris is circular and characterized by a large contrast to the sclera. Therefore, it seems obvious to use a contour based tracker. Witzner et al. [33] describe an algorithm for tracking using active contours and particle filtering. A generative model is formulated which combines a dynamic model of state propagation and an observation model relating the contours to the image data. The current state is then found recursively by taking the sample mean of the estimated posterior probability.

The proposed method is based on [33], but extended with constraints and robust statistics.

4.3.1 The dynamic model

The dynamic model describes how the iris moves from frame to frame. Again, the iris is modeled as an ellipse and the state vector x consist of the five parameters defining an ellipse as defined in equation 4.3.

To define the problem of tracking, consider the evolution of the state sequence

$$\mathrm{x}_{t+1} = \mathrm{f}_{t+1}\{\mathrm{x}_t, t \in \mathbf{N}\}, \quad (4.5)$$

of a target, given by

$$x_{t+1} = f_{t+1}(x_t, v_t), \qquad (4.6)$$

where f_{t+1} is a possibly non-linear function of the state x_t and $\{v_t, t \in N\}$ is an independent identically distributed process noise sequence. The objective of tracking is to recursively estimate x_{t+i} from the measurements,

$$M_{t+1} = h_{t+1}(x_{t+1}, n_{t+1}), \qquad (4.7)$$

where h_{t+i} is a possibly non-linear function and $\{n_{t+1}, t \in N\}$ is an i.i.d measurement noise sequence.

The pupil movements can be very rapid and is therefore modeled as Brownian motions (AR(l)). Thus the evolution of the state sequence (4.6) is modeled,

$$x_{t+1} = x_t + v_t, \quad v_t \sim N(0, \Sigma_t), \qquad (4.8)$$

where Σ_t is the time dependent covariance matrix of the noise. The time dependency compensates for scale changes, which affects the amount of movement. Larger movements is expected when the ellipse appears large, since the position of the eye is nearer to the camera. Contrary, when the eye is farther from the camera, smaller movements are expected. Hence, the first two diagonal elements of Σ_t corresponding to c_x and c_y are assumed to be linear dependent on previous sample mean.

4.3.2 The observation model

The observation model consists of two parts; a geometric component defining a probability density function over image locations of contours and a texture component defining a pdf over pixel gray level differences given a contour location. The geometric component models the deformations of the iris by assuming Gaussian distribution of all sample points along the contour. The gray level information is gathered by sampling a discrete set of points along the normals of all contour sampling points. Both components are joined and marginalized to produce a test of the hypothesis that there is a true contour present. The contour maximizing the combined hypotheses is chosen, see [33] for details.

4.3.3 Active contour tracking

The probabilistic formulation has the attraction that uncertainty is handled in a systematic fashion - Increased uncertainty results the particles to be drawn from a wider distribution, while increased confidence results the particles to be drawn from a narrower distribution.

The prediction stage involves using the system model (4.6) to obtain the prior pdf of the state at time t+1,

$$p(x_{t+1} \mid M_t) = \int p(x_{t+1} \mid x_t) p(x_t \mid M_t) dx_t \qquad (4.9)$$

The observation Mt is independent of the previous state x_{t-1} and previous observation M_{t-1} given the current state x_t. At time step t+1 a measurement M_{t+1} becomes available. This is used to update the prior via Bayes' rule,

$$p(x_{t+1} \mid M_{t+1}) \propto p(M_{t+1} \mid x_t) p(x_{t+1} \mid M_t). \qquad (4.10)$$

With this in mind, the tracking problem is stated as a Bayesian inference problem by use of (4.9) and (4.10). Particle filtering is used with the purpose to estimate the filtering distribution $p(x_t|M_t)$ recursively. This is done through a random weighted sample set $S_t^N = \{(x_t^n, \pi_t^n)\}$, where n is the n^{th} sample of a state at time t weighted by π_t^n. The samples are drawn from the prediction prior distribution $p(x_{t+1}|M_t)$. The samples are weighted proportionally to the observation likelihood $p(x_t|M_t)$ given by the contour hypotheses. This sample

set propagates into a new sample set S_{t+1}^{N}, which represents the posterior probability distribution function $p(x_{t+1}|M_{t+1})$ at time t+1.

4.3.4 Constraining the hypotheses

Corneal reflections may confuse the algorithm to weigh some of the hypotheses unreasonably high compared to others. This issue is illustrated left in Figure 4.5, where the relative normalized weighting is colored in a temperature scale - Blue indicates low, while red high scores. By using robust statistics, these hypotheses are treated as outliers and therefore rejected.

The contour algorithm may fit to the sclera rather than the iris. This is due to the general formulation of absolute gray level differences ΔM [34], which seeks to detect contours in a general sense. An example is depicted in Figure 4.6, where the image evidence of the contour surrounding the sclera is greater than the one around the iris. It turns out that for a large number of particles, the maximum likelihood estimate prefers the contour around the white sclera when the gaze is turned towards the sides.

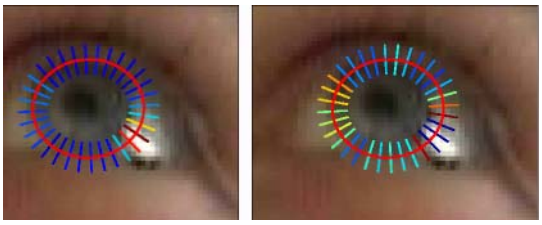


Figure 4.5: The relative normalized weighting of the hypotheses regarding one particle are colored in a temperature scale - Blue indicates low, while red high scores. *(Left)* Corneal reflections cause very distinct edges. Thus some hypotheses are weighted unreasonable high, which may confuse the algorithm. *(Right)* By use of robust statistics outliers are rejected. This results in a better and more robust estimate of the hypotheses regarding the contour.

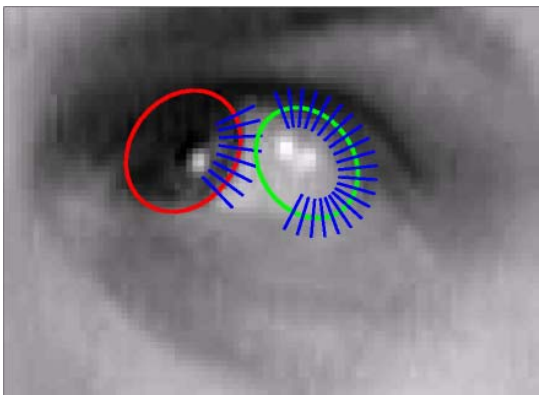


Figure 4.6: This figure illustrates the importance of the gray level constraint. Due to the general formulation of absolute gray level differences, the right contour has a greater likelihood, and the algorithm may thus fit to the sclera. Note the low contrast between iris and skin.

As a consequence, we propose to constrain the hypotheses. Intuitively, the average intensity value of the inner ellipse could be compared to some defined outer region as seen in expression (4.2). This is a poor constraint due to corneal reflection causing white blobs in the pupil area. The robustness of the active contour algorithm is increased by weighing the belief of hypotheses and utilizing robust statistics to reject outliers. We propose to weigh the hypotheses through a sigmoid function, applied on the measurement line M, defined as,

$$W = \left(1 + \exp\left(\frac{\mu_i - \mu_0}{\sigma_w}\right)\right)^{-1} \qquad (4.11)$$

where σ_w adjust the slope of weighting function, μ_i and μ_0 are the mean values of the inner and outer sides of the contour respectively. The function is exemplified in Figure 4.7. This has the effect of decreasing the evidence when the inner part of the ellipse is brighter than the surroundings. In addition, this relaxes the importance of the hypotheses along the contour around the eyelids, which improves the fit.

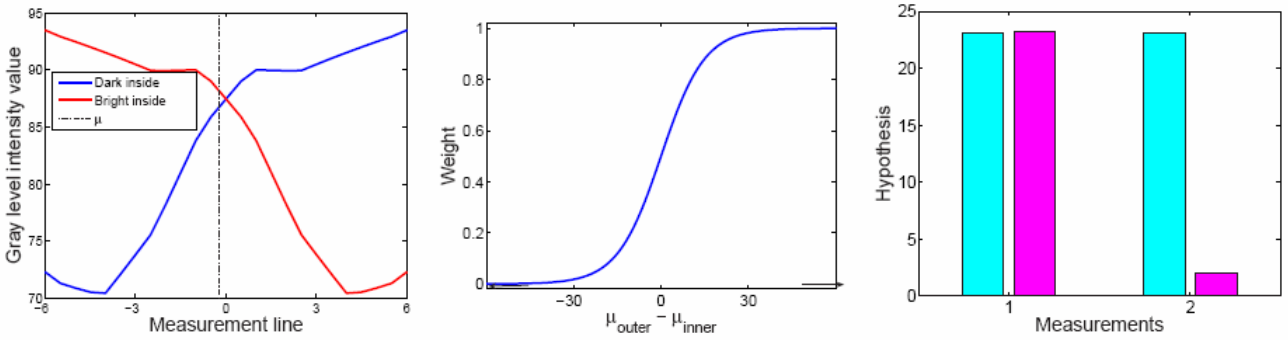


Figure 4.7: *(Left)* The two lines depict the gray level intensity of two measurement lines - The blue one where the inner part of the ellipse is dark, and the red in the reverse case. *(Middle)* The shifted hyperbolic tangents are utilized as weighting function. Note, the limit values are in range [—255; 255]. *(Right)* The cyan bars indicate the hypothesis value before weighting, while the pink is after. Measurement 1 - The blue line - is nearly unchanged, while 2 - the red line - is suppressed.

4.3.5 Maximum a posteriori formulation

The dynamic model may, in certain outlier cases, grow or shrink the contour to a degree, from where the algorithm gets lost. As a consequence, we propose to constrain on the shape of the ellipse in analogy to section 4.2. The parameters defining an ellipse are assumed normally distributed with mean μ and covariance $\mathbf{\Sigma}$. The prior distribution of these parameters is then defined,

$$p(\mathbf{x}) = N(\mu, \mathbf{\Sigma}) \propto \exp\left(-\frac{1}{2}(\mathrm{x} - \mu)^T \mathbf{\Sigma}^{-1}(\mathrm{x} - \mu)\right), \qquad (4.12)$$

where the normalization factor has been omitted. The mean and covariance are estimated in a training sequence.

Combining the priors – presented in this section – with the likelihood, results in the *Maximum a Posteriori* formulation (MAP), where the goal is to maximize,

$$p(\mathrm{x} \mid M) \propto p(M \mid \mathrm{x}) p(\mathrm{x}) \qquad (4.13)$$

By incorporation of prior knowledge about the shape, with the prediction prior and observation likelihood (10), the robustness increases considerably and the parameters are constrained to avoid the algorithm to break down due to infinite increase or decrease of parameters.

4.3.6 Results

A number of experiments have been performed with the proposed methods. We wish to investigate the importance of image resolution. Therefore the algorithms are evaluated on two datasets. One containing close up images, and one containing a down-sampled version hereof.

The algorithms estimate the center of the pupil. For each frame the error is recorded as the difference between a hand annotated ground truth and the output of the algorithms. This may lead to a biased result due to annotation error. However, this bias applies to all algorithms and a fair comparison can still be made.

Figure 4.8 and 4.9 depicts the error as a function of the number of particles used, for low resolution and high resolution images respectively. The errors for three different active contour (AC) algorithms are shows; basic, with EM refinement, and finally with deformable template (DT) refinement. The error of the deformable template (DT) algorithm, initialized by double threshold, is inserted into the plot.

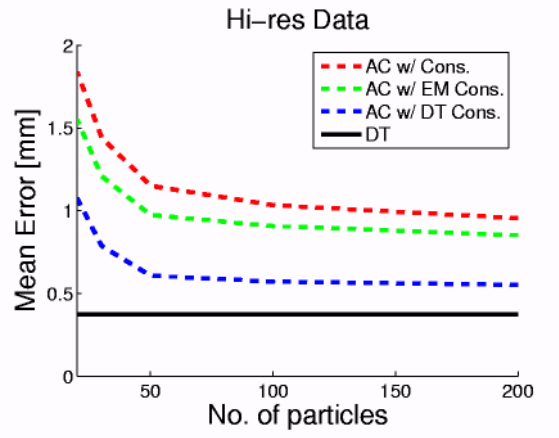


Figure 4.8: The error of the algorithms as a function of the number of particles for the high resolution data.

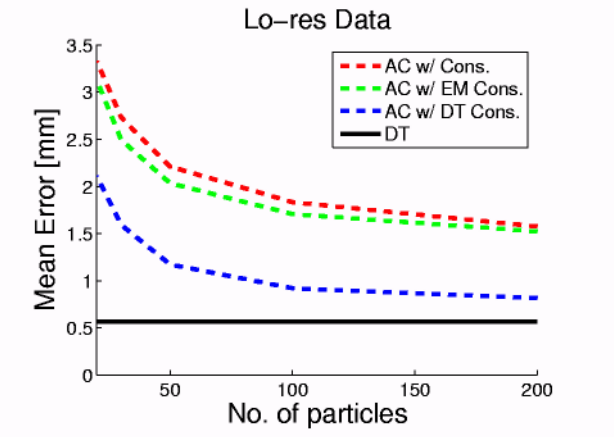


Figure 4.9: The error of the algorithms as a function of the number of particles for the low resolution data.

Hi-res	*E(x,y)* [mm]	*E(θ)*	[frame/s]
AC	0.9	4.1	0.54
AC w/EM	0.8	3.7	0.49
AC w/DT	0.5	2.3	0.25
DT	0.3	1.4	2.2
Lo-res			
AC	1.5	7.3	0.57
AC w/EM	1.5	6.9	0.55
AC w/DT	0.8	3.7	0.49
DT	0.5	2.3	8.4

Table 4.1. Speed and precision comparison of the algorithms. The active contour uses 200 particles.

It can be seen that the proposed constraints on the active contour generally improves the accuracy of the fit. The refinement by the deformable template performs better than the EM method. The cost is an increased number of computations, which is resolution dependent. Nonetheless, the deformable template method, initialized by double thresholding, is seen to outperform all active contour algorithms.

The table in figure 4.4 lists the mean error in accuracy in centimeters and degrees. Also listed is the computation time in frames per section of a Matlab implementation run on a 2.4 GHz PC. In general, the accuracy improves with high resolution as seen in table 4.1. However, the methods utilizing deformable template matching are less sensitive. The computation time for the basic active contour and EM refinement methods are independent of resolution. A significant increase in speed is noticed for the deformable template methods.

4.3.7 Conclusion

Here we have presented heuristics for improvement of the active contour method proposed by [33]. We have shown increased performance by using the prior knowledge that the iris is darker than its surroundings. This prevents the algorithm from fitting to the sclera as seen in Figure 4.6.

Also presented is a novel approach to eye tracking based on a deformable template initialized by a simple heuristic. This enables the algorithm to overcome rapid eye movements. The active contour method handles these by broadening the state distribution and thus recovering the fit in a few frames. Furthermore, the accuracy is increased by fitting to the pupil rather than iris. This is particularly the case when a part of the iris is occluded as seen in Figure 4-10. It is shown that the deformable template model is accurate independent of resolution and it is very fast for low resolution images. This makes it useful for head pose independent eye tracking.

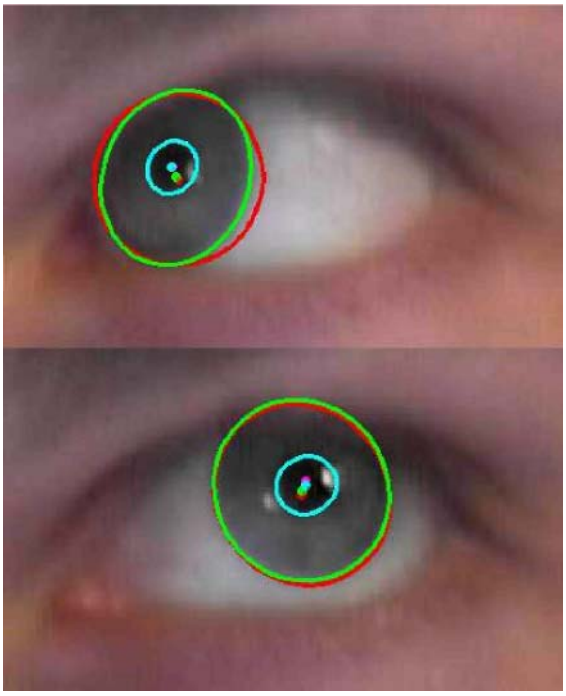


Figure 4.10: The resulting fit on two frames from a sequence - the red contour indicates the basic active contour, green indicates the EM refinement and the cyan indicates the deformable template initialized by the heuristic method. The top figure illustrates the benefit fitting to the pupil rather than the iris. Using robust statistic the influences from corneal reflections on the deformable template fit are ignored as depicted in the bottom image.

5 Head Orientation Estimation[7]

5.1 Head Modeling Using Active Appearance Modeling

Active appearance models combine information about shape and texture. In [35] *shape* is defined as "...that quality of a configuration of points which is invariant under some transformation." Here a face shape consists of ***n*** 2D points, *landmarks,* spanning a 2D mesh over the object in question. The landmarks are either placed in the images automatically [36] or by hand. Figure 5.1 shows an image of a face [57] with the annotated shape shown as a red dots. Mathematically the shape **s** is defined as the 2n-dimensional vector of coordinates of the n landmarks making up the mesh,

$$s = \left[x_1, x_2, \ldots, x_n, y_1, y_2, \ldots, y_n\right]^T \qquad (5.1)$$

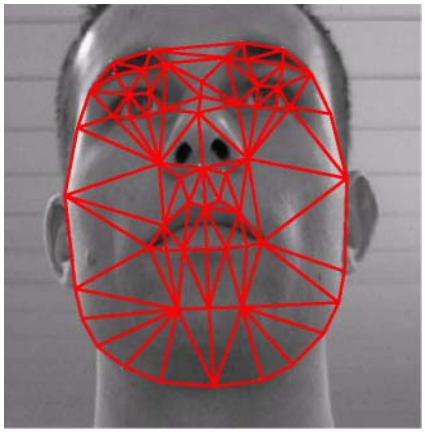


Figure 5.1: Face image of a face annotated with 58 landmarks for active appearance modeling.

Given *N* annotated training examples, we have *N* such shapevectors **s**, all subject to some transformation. In 2D the transformations considered are the similarity transformations (rotation, scaling and translation). We wish to obtain a model describing the inter-shape relations between the examples, and thus we must remove the variation given by this transformation. This is done by aligning the shapes in a common coordinate frame as described in the next section.

To remove the transformation, ie. the rotation, scaling and translation of the annotated shapes, they are aligned using iterative Procrustes analysis [35]. Figure 5.2 shows the steps of the iterative Procrustes analysis. The top figure shows all the landmarks of all the shapes plotted on top of each other. The center figure shows the initialization of the shape by the translation of their centers of mass and normalization of the norm of the shape vectors. The right figure is the result of the iterative Procrustes algorithm.

The normalization of the shapes and the following Procrustes alignment results in the shapes lying on a unit hypersphere. Thus the shape statistics will have to be calculated on the surface of this sphere. To overcome

[7] Chapter 5 is based on a draft manuscript; the final paper was published as:

Vester-Christensen, M., Leimberg, D., Ersbøll, B.K., Hansen, L.K. (2005) Towards emotion modeling based on gaze dynamics in generic interfaces. *Proceedings of HCI International 2005,* published on CD-ROM by Lawrence Erlbaum Associates, Inc (ISBN 0-8058-5807-5).

this problem the approximation that the shapes lie on the tangent plane to the hypersphere is made, and ordinary statistics can be used. The shape **s** can be projected onto the tangent plane using:

$$\mathbf{s}' = \frac{\mathbf{s}}{\mathbf{s}^T\mathbf{s}_0} \qquad (5.2)$$

where **s** is the estimated mean shape given from the Procrustes alignment.

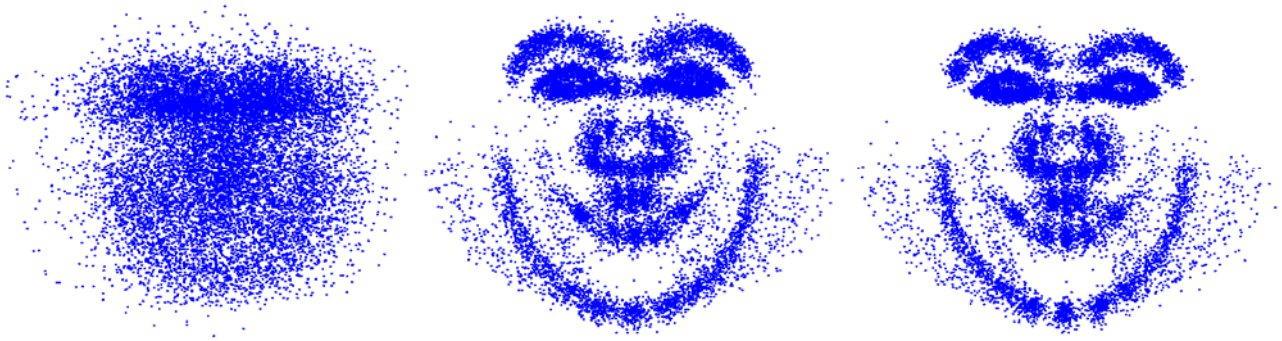


Figure 5.2: Procrustes analysis. The left figure shows all landmark points plotted on top of each other. The center figure shows the shapes after translation of their centers of mass, and normalization of the vector norm. The right figure is the result of the iterative Procrustes alignment algorithm.

With the shapes aligned in a common coordinate frame it is now possible to build a statistical model of the shape variation in the training set.

The result of the Procrustes alignment is a set of 2n dimensional shape vectors $\mathbf{s_i}$ forming a distribution in the space in which they live. In order to generate shapes, a parameterized model of this distribution is needed. Such a model is of the form $\mathbf{s} = M(\mathbf{b})$, where **b** is a vector of parameters of the model. If the distribution of parameters p(**b**) can be modeled, constraints can be put on them such that the generated shapes **s** are similar to that of the training set. With a model it is also possible to calculate the probability p(**s**) of a new shape.

To constitute a shape, neighboring landmark points must move together in some fashion. Thus some of the landmark points are correlated and the true dimensionality may be much less than 2n. Principal Component Analysis(PCA) rotates the 2n dimensional data cloud that constitutes the training shapes. It maximizes the variance and gives the main axis of the data cloud.

The PCA is performed as an eigenanalysis of the covariance matrix, **Σ**, of the training data.

$$\mathbf{\Sigma} = \frac{1}{N-1}\mathbf{S}\mathbf{S}^T \qquad (5.3)$$

where N is the number of training shapes, and **S** is the n x N matrix $\mathbf{D} = [\mathbf{s_1}\text{-}\mathbf{s_0}, \mathbf{s_2}\text{-}\mathbf{s_0}, \ldots, \mathbf{s_n}\text{-}\mathbf{s_0}]$. **Σ** is an n x n matrix. Eigenanalysis of the **Σ** matrix gives a diagonal matrix $\mathbf{\Lambda_I}$ of eigenvalues λ_i and a matrix $\mathbf{\Phi_I}$ with eigenvectors $\boldsymbol{\varphi}_\mathbf{i}$ as columns. The eigenvalues are equal to the variance in the eigenvector direction.

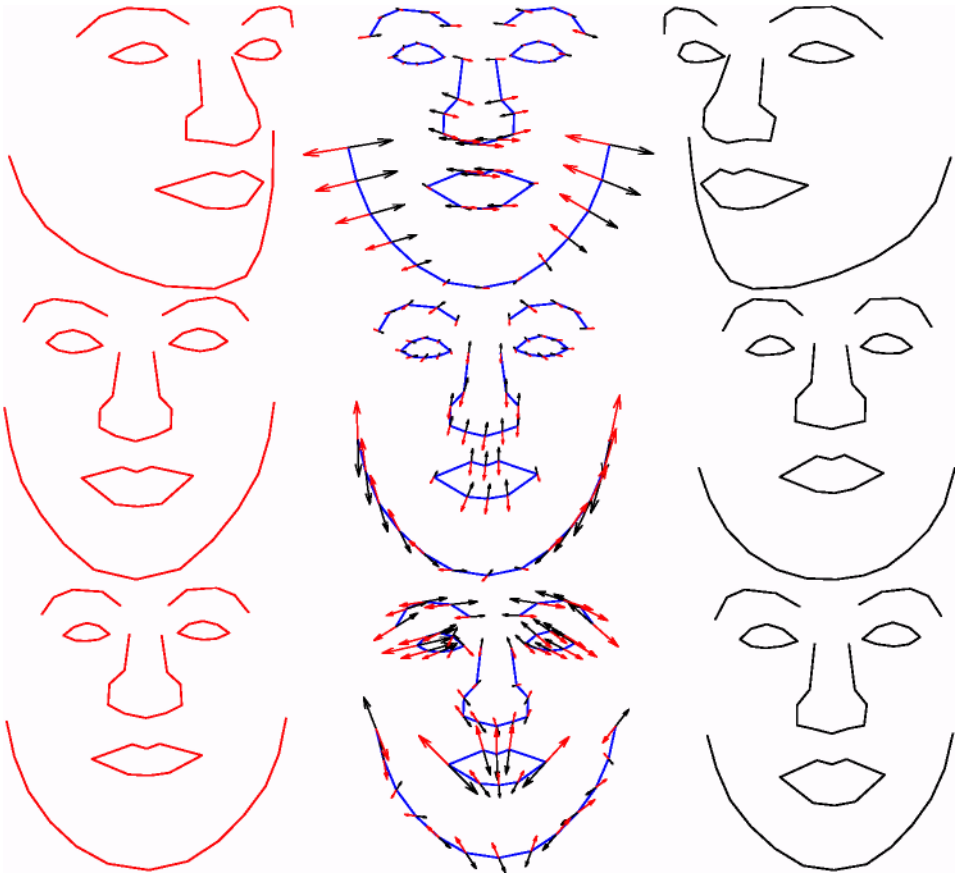


Figure 5.3: Mean shape deformation using first, second and third principal mode. The middle shape is the mean shape, the left column is minus two standard deviations corresponding to $b_{si} = -2\lambda$, the right is plus two standard deviations given by $b_{si} = 2\lambda$. The arrows overlain the mean shape indicates the direction and magnitude of the deformation corresponding to the parameter values.

PCA can be used as a dimensionality reduction tool by projecting the data onto a subspace which fulfills certain requirements, for instance retaining 95% of the total variance or similar. Then only the eigenvectors corresponding to the t largest eigenvalues fulfilling the requirements are retained. This enables us to approximate a training shape instance **s** as a deformation of the mean shape by a linear combination of t shape eigenvectors,

$$\mathbf{s} \approx \mathbf{s}_0 + \mathbf{\Phi}_s \mathbf{b}_s \qquad (5.4)$$

where $\mathbf{b}_s$ is a vector of t *shape parameters* given by

$$\mathbf{b}_s = \mathbf{\Phi}_s^T (\mathbf{s} - \mathbf{s}_0) \qquad (5.5)$$

and $\mathbf{\Phi}_s$ is the matrix with the t largest eigenvectors as columns.

A synthetic shape **s** is created as deformation of the mean shape $\mathbf{s}_0$ by a linear combination of the shape eigenvectors $\mathbf{\Phi}_s$.

$$\mathbf{s} = \mathbf{s}_0 + \mathbf{\Phi}_s \mathbf{b}_s, \qquad (5.6)$$

where bs is the set of shape parameters. Specific facial expression may be learnt from examples in the AAM representation and re-synthesized by AAM simulation. Figure 5.3 shows three rows of shapes indicating the

flexibility of the representation. The middle row is the mean shape. The left and right rows are synthesized shapes generated by deformation of the mean shape by $\pm 2\sqrt{\lambda_i}$.

In order to track moving faces, the AAM must be re-estimated for each frame. The objective is then to find the optimal set of parameters $\mathbf{b_s}$ and $\mathbf{b_g}$ such that the model instance $T(\mathbf{W}(\mathbf{x},\mathbf{b_s}))$ is as similar as possible to the object in the image. An obvious way to measure the success of the fit is to calculate the error between the image and the model instance. An efficient way to calculate this error is to use the coordinate frame defined by the mean shape $\mathbf{s_0}$. Thus a pixel with coordinate $\mathbf{x}$ in $\mathbf{s_0}$ has a corresponding pixel in the image $\mathbf{I}$ with coordinate $\mathbf{W(x,b_s)}$ as described previously. The error of the fit can then be calculated as the difference in pixel values of the model instance and the image:

$$\mathbf{f}(\mathbf{b_s},\mathbf{b_g}) = (\mathbf{g_0} + \mathbf{\Phi_g}\mathbf{b_g}) - \mathbf{I}(\mathbf{W}(\mathbf{x},\mathbf{b_s})), \qquad (5.7)$$

This is a function in the texture parameters $\mathbf{b_g}$ and the shape parameters $\mathbf{b_s}$. A cost function can be defined as,

$$\mathbf{F}(\mathbf{b_s},\mathbf{b_g}) = \left\| \mathbf{g_0} + \mathbf{\Phi_g}\mathbf{b_g} - \mathbf{I}(\mathbf{W}(\mathbf{x},\mathbf{b_s})) \right\|^2 \qquad (5.8)$$

The optimal solution to (8) can be found as,

$$(\mathbf{b_s^*},\mathbf{b_g^*}) = \arg\min_{\mathbf{b_s},\mathbf{b_g}} \mathbf{F}. \qquad (5.9)$$

Solving this, is in general a non-linear least squares problem, but fortunately there exist well-proven algorithms [38] for this step. The optimal shape can then be used for posture estimation.

5.2 Estimation of Head Orientation Using Characteristic Points of Face

5.2.1 Introduction

Head pose determination represents an important area of research in human computer interaction (HCI). There are many researches in the area of estimation with monocular vision [39]. Methods for head estimation can be classified into two main categories: model based and face property-based. Model-based use 3D model of the face and typically recover the face pose by first establishing 2-3D features correspondences and then solving for the face pose using the conventional pose estimation techniques. Property-based approaches, assume there exists a unique causal-effect relationship between 3D face pose and certain properties of the facial image. Their goal is to determine the relationship from a large number of training images with known 3D face poses. They use neural networks to construct a mapping between 2D face images and 3D face poses [40], but, it cannot work well for previously unseen persons and backgrounds. Other proposed to use eigenspace for face detection and head pose estimation [41]. In summary, the property-based methods are simpler, but less accurate, many require a large number of training face image under different orientations.

Model-based methods usually start with feature detection, followed by matching 2D/3D corresponding features and determining face pose using the matched features. Among all facial features, the most commonly used are eyes, nose, mouth [42, 43].

In view this solution, we propose model-based method for estimation 3D head position. First, we created head model, where 3D rotation centre is located between eyes (Figure 5.5). Second, the algorithm for characteristic point of face detection is discussed. Third, the algorithm of detection of head rotation angles from the found points is discussed.

5.2.2 Head 3D pose estimation method

Head has six degree of freedom: three rotation angles (Figure 5.4) and three linear shifts. In general, a head rotation may be characterized by three Euler angles roll, pitch and yaw (Figure 5.4). Head 3D rotation matrixes **R** can be obtained from three single Euler rotations: angle around *z* (roll, $R_{z,\varphi,}$) next around y (yaw, $R_{y,\Theta,}$) and finally around *x* (pitch, $R_{x,\psi}$):

$$R_{z,\varphi} = \begin{bmatrix} \cos\varphi & -\sin\varphi & 0 \\ \sin\varphi & \cos\varphi & 0 \\ 0 & 0 & 1 \end{bmatrix};$$
$$R_{y,\Theta} = \begin{bmatrix} \cos\Theta & 0 & \sin\Theta \\ 0 & 1 & 0 \\ -\sin\Theta & 0 & \cos\Theta \end{bmatrix}; \qquad (5.10)$$
$$R_{x,\psi} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\psi & -\sin\psi \\ 0 & \sin\psi & \cos\psi \end{bmatrix}.$$

Thus head 3D rotation matrixes **R** can be expressed:

$$\mathbf{R}=\begin{bmatrix} c\varphi\cdot c\Theta & c\varphi\cdot s\Theta\cdot s\psi - s\varphi\cdot c\psi & c\varphi\cdot s\Theta\cdot c\psi + s\varphi\cdot s\psi \\ s\varphi\cdot c\Theta & s\varphi\cdot s\Theta\cdot s\psi + c\varphi\cdot c\psi & s\varphi\cdot s\Theta\cdot c\psi - c\varphi\cdot s\psi \\ -s\Theta & c\Theta\cdot s\psi & c\Theta\cdot c\psi \end{bmatrix} \qquad (5.11),$$

here "s" is sin ; "c" is cos.

Local reference point and 3D rotation centre of head $\{X_0, Y_0, Z_0\}$ centre is located between eyes. New local coordinate is recalculated from eyes coordinate (Z_0=0):

$$X_0 = X_l + \frac{X_r - X_l}{2};\ Y_0 = Y_l + \frac{Y_r - Y_l}{2} \qquad (5.12)$$

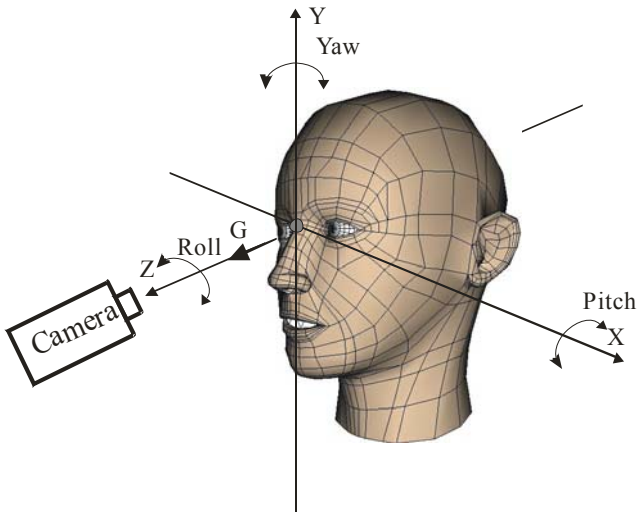


Figure 5.4: The definition of three head rotation angles: roll, yaw, pitch and camera position.

Head's ground-start position is describing over direction angle *G*, which is parallel to camera optical axis (Figure 5.4). Coordinates are: $\{X_{pl}, Y_{pl}, Z_{p}\}$ – left eye, $\{X_{pr}, Y_{pr}, Z_{pr}\}$ – right eye and $\{X_{pn}, Y_{pn}, Z_{pn}\}$ – nose. After rotation characteristic points coordinates are $\{X_{fl}, Y_{fl}, Z_{fl}\}$, $\{X_{fr}, Y_{fr}, Z_{fr}\}$, $\{X_{fn}, Y_{fn}, Z_{fn}\}$. Relation between ground-start position and final head position can be described by matrix **R:**

$$\mathbf{R} \cdot \begin{bmatrix} X_{px} \\ Y_{px} \\ Z_{px} \end{bmatrix} = \begin{bmatrix} X_{fx} \\ Y_{fx} \\ Z_{fx} \end{bmatrix}, \qquad (5.13)$$

here and forward $x \in (l$ – coordinate left eye, r – coordinate right eye, n – coordinate nose$)$.

After rotation the between head and camera have shifting. Because of this reason we calculated head drift coefficient *d* by equation:

$$d = \frac{\sqrt{(x_{pr} - x_{pl})^2 + (y_{pr} - y_{pl})^2}}{X_E}, \qquad (5.14)$$

here X_E – parameter, described in next section.

All coordinate are multiplied by *d*. Using obtained parameter *d* we can rewrite (5.13) as:

$$\mathbf{R} \cdot \begin{bmatrix} X_{px} \cdot d \\ Y_{px} \cdot d \\ Z_{px} \cdot d \end{bmatrix} = \begin{bmatrix} X_{fx} \cdot d \\ Y_{fx} \cdot d \\ Z_{fx} \cdot d \end{bmatrix}. \qquad (5.15)$$

If coordinates define over vector $\mathbf{v}_p = (X_{px} \cdot d, Y_{px} \cdot d, Z_{px} \cdot d)^T$ and , $\mathbf{v}_f = (X_{fx} \cdot d, Y_{fx} \cdot d, Z_{fx} \cdot d)^T$. There are naturally got nine equations, but we are eliminated equations with final positions Z_{fl}, Z_{fr}, Z_{fn} – deepness coordinates, which are impossible get from final image. Equation (6) can be rewrite as:

$$\varepsilon_1 = \sum_{j=1}^{3} \mathbf{R}_{1,j} \cdot \mathbf{v}_{pj}^{(i)} - \mathbf{v}_{f1}^{(i)} \; ; \varepsilon_2 = \sum_{j=1}^{3} \mathbf{R}_{1,j} \cdot \mathbf{v}_{pj}^{(i)} - \mathbf{v}_{f2}^{(i)}, \qquad (5.16)$$

where i=1..3.

System of six equations can be solved using minimization least squares method:

$$E = \sum \varepsilon_1^2 + \sum \varepsilon_2^2 \qquad (5.17)$$

After minimization least squares method (8) we obtained tree angles φ, Θ, ψ, whish characterized head 3D rotation.

5.2.3 Head model

To use our algorithm, we need to measure few face parameters and to create head model. It's start procedure of our method.

We selected next parameters: Z and Y coordinate of nose tip $Z_{n,}$, Y_n, depth of eyes - Z_l, Z_r ($Z_l = Z_r$) and distance between eyes - X_E (Figure 5.5).

There were analysed about 300 different head pictures. The needed parameters were evaluated statistically: $Z_n = 21$ mm, $Y_n = 41$ mm, $Z_{l,r} = 19.5$ mm, $X_E = 70$ mm.

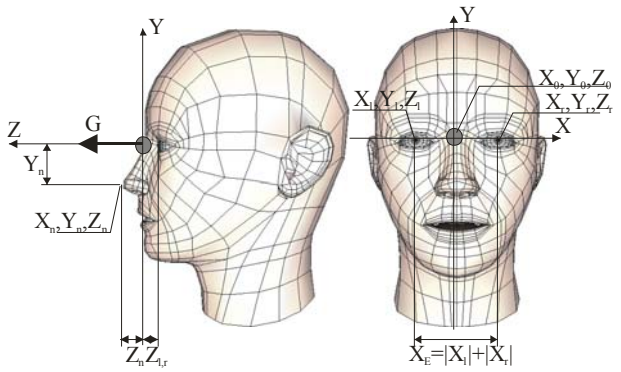


Figure 5.5: Location of characteristics points and centre of rotation.

5.2.4 Face detection and tracking

The quality of image is important for good detection and tracking. Our aim is create method, which works with usual web camera. We use PC web camera (possible resolutions 240x120, 320x240, 640x480) and 24 bit colour depth. Next important criterion is good and smooth lighting. Face should fill 60-90% of view.

We used several methods to extract face and characteristic points (eyes, nose, lips): extracting characteristic face points from 2D image created by author [44], detection by skin colour [45] and OpenCV's rapid object detection [46, 47]. Fastest and best results showed OpenCV. For object detection OpenCV require training with positive and negative objects (eyes, nose) samples. OpenCV was training with 1500 positive and 5000 negative sample. Samples where collected from several database: positive from "Color FERET"[48] and "Cohn–Kanade AU–Coded Facial Expression Database"[49], negative - CorelDraw PhotoCD. For better detection OpenCV was improved by additional facial features [5-16]. Method's summary accuracy of eyes detection is about 94-96% (detecting pair of eyes), nose about 94%. OpenCV is used as primary eyes and nose coordinates detector. For continuously real time tracking the normalized correlation method was used. If error occurred during tracking by correlation the points detected by OpenCV is repeated (Figure 5.6).

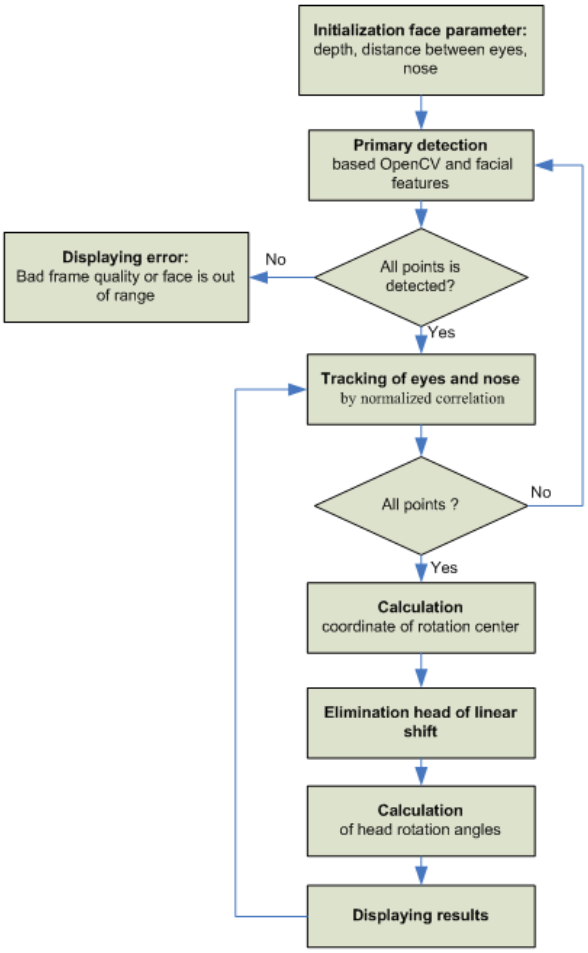


Figure 5.6: Proposed head pose estimation algorithm using characteristic points of face.

5.2.5 Results

To evaluate our algorithm, we used characteristic points coordinates generate by 3D head model. Finally coordinates are corrupted by Gausian noise with standard deviation 0.1, 0.5, 1.0 and 2.0. All rotation angles were equal to 10 degrees, and camera resolution was 240x120, 480x240 and 640x480. The estimated orientations were compared with calculated "ground-truth" orientation. Experimental results shown in Figure 5.7.

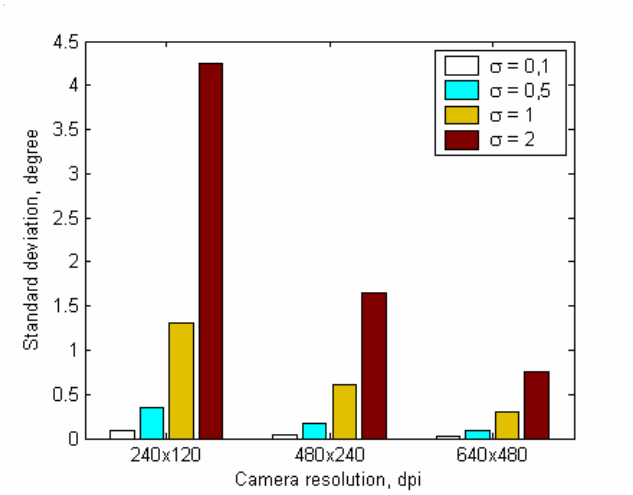


Figure 5.7 Accuracy of estimation head angle under different camera resolution and noise level (σ), where angles φ = 10°, Θ = 10°, ψ = 10°

In Figure 5.8 are shown angles estimation absolute errors versus different rotations (Yaw, Pitch and Roll). If one of angles is changed, another's are set on 10 degrees. Noise level σ=1, camera resolution 320x240. Remark that yaw angle less 0 degree is more unstable - absolute error approximately 1 degree, than more 0 degree absolute error is approximately 0.4 degree. Pitch and roll is stable in all range approximately 0.2 – 0.4 degree.

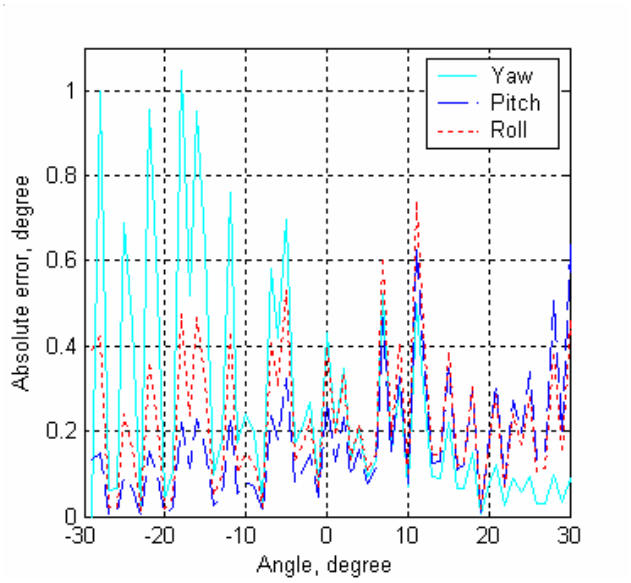


Figure 5.8. Angles estimation absolute error versus different simultaneous rotations (other angle equal 10 degree) under noise level σ=1, camera resolution 320x240 dpi

Often in HCI a roll of head is secondary angle. Are used only two rotation angles: pitch and yaw. In Figure 5.9 are shown head gaze direction angle error versus yaw and pitch rotations under roll φ=0°.

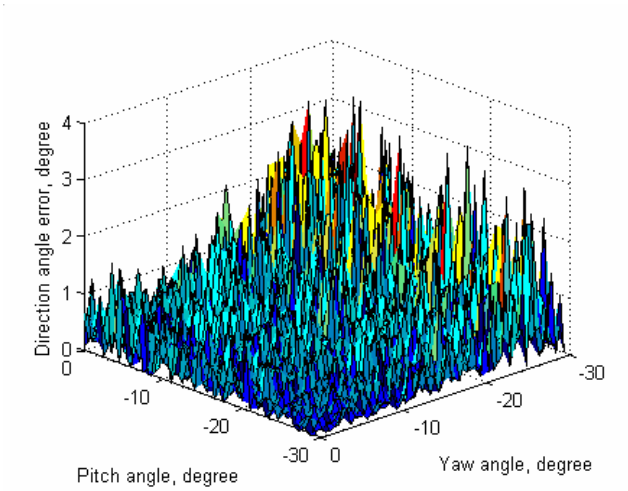


Figure 5.9: 3D plot of the direction angle errors versus yaw and pitch rotations, under roll = 0°, noise level σ = 1, camera resolution 320x240 dpi

5.2.6 Conclusion

We described 3D head pose estimation from a single monocular camera. Simulation of errors show, that our method's sensitivity is enough high. Without noise under normal resolution (320x240) the estimation error are less 0.3 degree, with noise level σ=1, error less 1 degree.

Very important is high accuracy and robustness of characteristic point detection. We neglected to face expression and it must be further research direction.

6 Examples of Practical Implementations

6.1 Single-Camera Remote Eye Tracker

A prototype of a remote eye tracker using a single high-resolution camera and two infra-red light sources has been developed at the Institute for Neuro- and Bioinformatics of the University of Lübeck (UzL) [50, 51].

Figure 6.1: System setup of the remote eye tracker. The tracker consists of a single high-resolution camera and two infra-red light sources to either side of the camera, mounted below a computer display.

The tracker allows head movements in a volume of around 20x20x20cm and achieves an accuracy of around 1.5 degrees; we hope to improve this further by fine-tuning the image processing. The sampling rate is 15 Hz, and we hope to increase this to around 50 Hz in the near future. The tracker was demonstrated at the PIT 2006 workshop and at the COGAIN camp.

The eye tracking hardware, shown in Figure 6.1, consists of a single high-resolution industrial camera and two infra-red light sources mounted below a computer display. The infra-red light sources generate reflections (corneal reflexes, CRs) on the surface of the cornea, which are used to locate the eyes in the camera image and determine the position of the eyes in space.

The eye tracker software consists of two main components: An image processing component, which determines the position of the CRs and pupils in the camera image, and a gaze estimation component, which uses this information to compute the direction of gaze.

Table 6.1 shows the result of accuracy measurements performed on the eye tracker with four test subjects. Each subject calibrated the eye tracker with their head in the centre of the working range, and eye tracking accuracy was then measured in several different head positions. The average accuracy was 1.57 degrees.

Head position	**(x, y, z)**	**RMS gaze error (degrees)**
centre (calibration position)	(0, 0, 0)	1.01
left	(-10, 0, 0)	1.33
top	(0, 10, 0)	1.23
top left	(-10, 10, 0)	1.44
front	(0, 0, -10)	2.54
back	(0, 0, 10)	1.88
Overall		**1.57**

Table 6.1: Accuracy measurements performed on the remote eye tracker for four test subjects. Root mean square (RMS) gaze error was measured in different head positions; coordinates (in centimetres) are given relative to the centre of the working range, which was at a distance of 58 cm from the screen.

6.2 Low Cost Eye Tracker

COGAIN partner UNI KO-LD (Universität Koblenz-Landau) developed GoldenGaze, a low cost, IR based eye tracking system, suitable to be used as an input device for systems like UKO-II.

Using of-the-shelf equipment at a total cost below € 150, an experimental system was developed to track gaze directions at a suitable precision to distinguish between 4 to 9 different directions. 4 directions are sufficient to control typing systems like UKO-II. The current implementation achieves a detection rate of 10 to 20 Hz at a resolution of approximately 4 degrees.

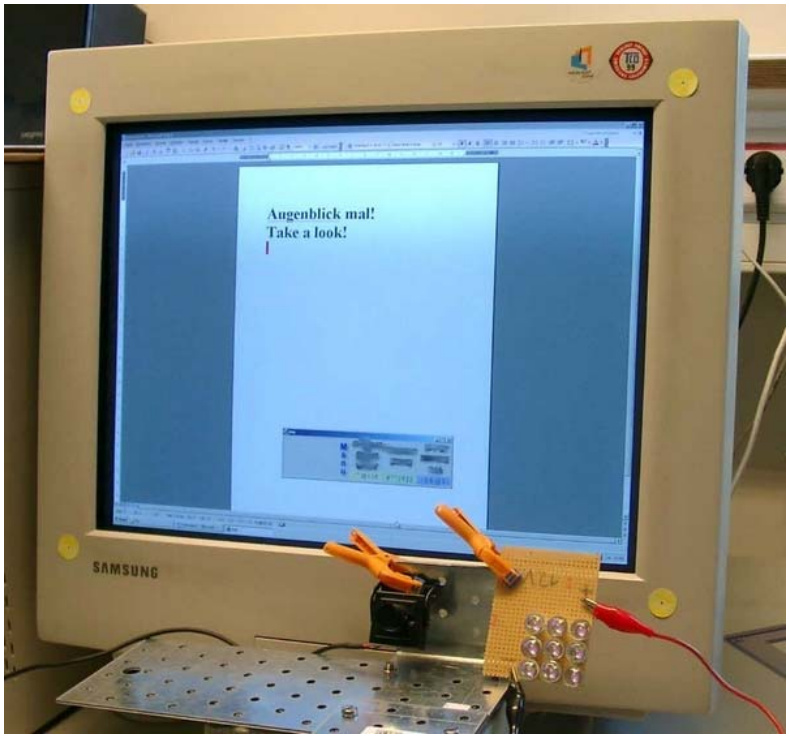


Figure 6.2: A GoldenGaze experimental setup, depicting the IR camera (black), the IR LEDs (right to the camera) and some eye-typed text on the screen.

The system uses the corneal-reflection-method with off-axis illumination. It tolerates head movements of +/- 15 cm in any direction and, to a certain degree, non-optimal illumination situations. Also, users may wear glasses or not.

Due to the low resolution signal of the camera (a high sensitivity analog monochrome surveillance camera at approx. € 80,-) the glint and pupil detection need to work on rather low resolution sub-images. This is the major reason for the rather low angle resolution of 4 degrees at present. Current work takes place in investigation and evaluation of better sub-pixel estimation algorithms for glint and pupil position estimation to improve the accuracy as well as in performance improvements to yield higher detection rates. The system works satisfying under lab condition, though the integration with UKO-II needs improvements. The first user test was promising, but showed the need for ergonomic changes to UKO-II to reflect the new interaction situation for the user.

6.3 I4Control® System

The I4Control® system [52, 53], developed in the Czech Technical University (CTU), uses a small camera mounted on a spectacles frame to monitor the user's eye position and movements. In this setting, the camera is automatically following the user's head and no extra control mechanism is necessary to ensure it. Moreover, this solution significantly simplifies identification of an eye in the camera image, which is rather simple and "standardized" (Figure 6.3) – the main difference among various images to be processed is the position of the iris and pupil. These properties decrease significantly requirements on image-processing as well as on the quantity of the input data (low cost CCD cameras can do for the purpose). This allows to drive down the overall price of the resulting device.

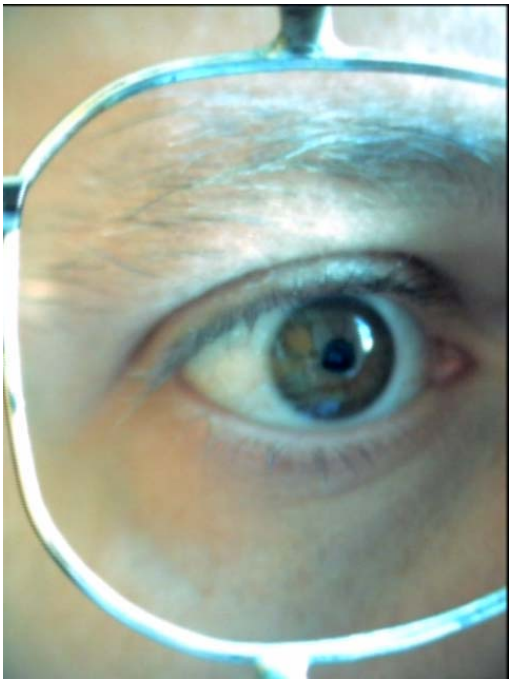


Figure 6.3: Original image of an eye as taken by the camera.

To ensure gaze interaction, data provided by a CCD camera has to be evaluated on-line. This can be ensured only if the system uses an evaluation algorithm, which is both robust and quick. In our first version of I4Control® system our aim was to identify the displacement of pupil form the basic balanced (or central) position. Pupil is identified as the darkest part of the processed image – pure thresholding on the level of individual pixels has been used in the detection algorithm. The threshold value was adjusted automatically to reduce the influence of illumination as much as possible. This straightforward and computationally simple technique results in very quick processing of the considered image. Unfortunately, its precision highly depends on the quality of illumination and the shades it eventually produces (e.g. that of nose, lashes, eye-brow, etc.). To reduce their impact the I4Control® detection algorithm has been enhanced so that its pre-processing phase complements the dynamic thresholding with a complementary condition requiring minimal number of dark points in direct neighbourhood of the considered interpreted pixel. In the next phase, the digital image resulting from this pre-processing is systematically searched for a sufficiently big dark compact spot. The first one found is identified as a pupil and its centre is reported as that of pupil. The resulting algorithm processes about 15 shots per second – if the processing of a certain shot is not ready in the corresponding time slot, this shot is neglected. Unfortunately, this solution is not robust enough. On the other hand it offers an important advantage from the user's point of view: it can be easily adjusted to an individual user.

The crucial part of any eye-tracker is its algorithm for identification of eye position, which has to be sufficiently robust. Illumination of the considered scene plays a decisive role here as much as in any other image processing task. If this problem is not taken care of correctly, the resulting product cannot identify the eye position correctly and consequently it gives erroneous results often. The first version of I4Control® passed the responsibility for proper illumination to the user. The second version, which is currently in the final stage of testing, introduces IR illumination of the scene.

IR illumination is introduced to improve robustness of the second version of I4Control® [54]. IR radiators are used as a complementary source of "light" – they are not switched on all the time but when the "natural" illumination is too weak. This solution tries to minimize possible health risks caused by long-term exposure to IR radiation (even though it is low and it meets the norm value set by the Czech law). Finding the

corresponding "norm value for IR radiation" proved to be rather demanding and time consuming and it caused certain delay in advance of the second version of I4Control®, which is now being prepared for production of a pilot series.

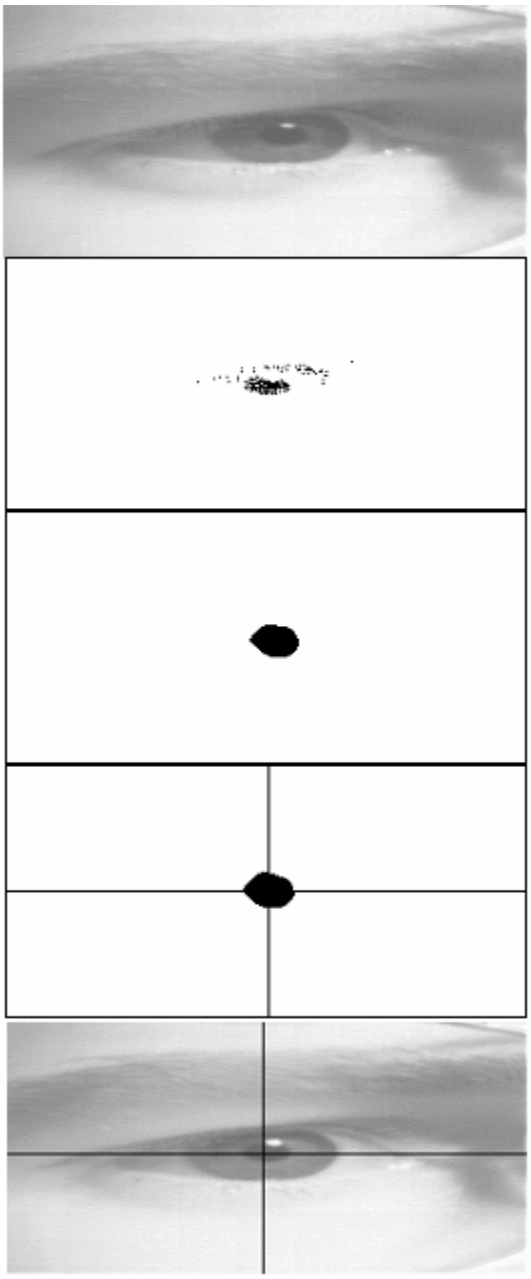


Figure 6.4: Image processing in the first version of I4Control® system.

The I4Control® system has to be individually calibrated for each user [55] – 2 zones have to be manually selected using classical PC mouse in the window of the calibration mode of the system. First, the individual *detection zone* is specified – see the larger blue rectangle in the Figure 6.5. It is the minimal part of the camera image, which fully covers the user's eye, i.e. the richest part concerning the gaze interaction – its position is influenced by the individual user's characteristics (his/her face anatomy, etc.). Second, the *balanced position* of the user's pupil has to be defined – it depicts a small part of the detection zone where the user's pupil is

placed when the user looks straight ahead. This very position (delimited by the smaller green rectangle in the Figure 6.4) is used as a reference point to which all the new data are compared –this comparison is used to control cursor movements:

- If the pupil centre is inside the balanced zone, the cursor stands still.
- In the other case the direction of cursor movement equals to the deviation of current position of the pupil centre from the balanced position. For example the situation on the Fig. 3 corresponds to the control signal "move the cursor in the north-west direction".

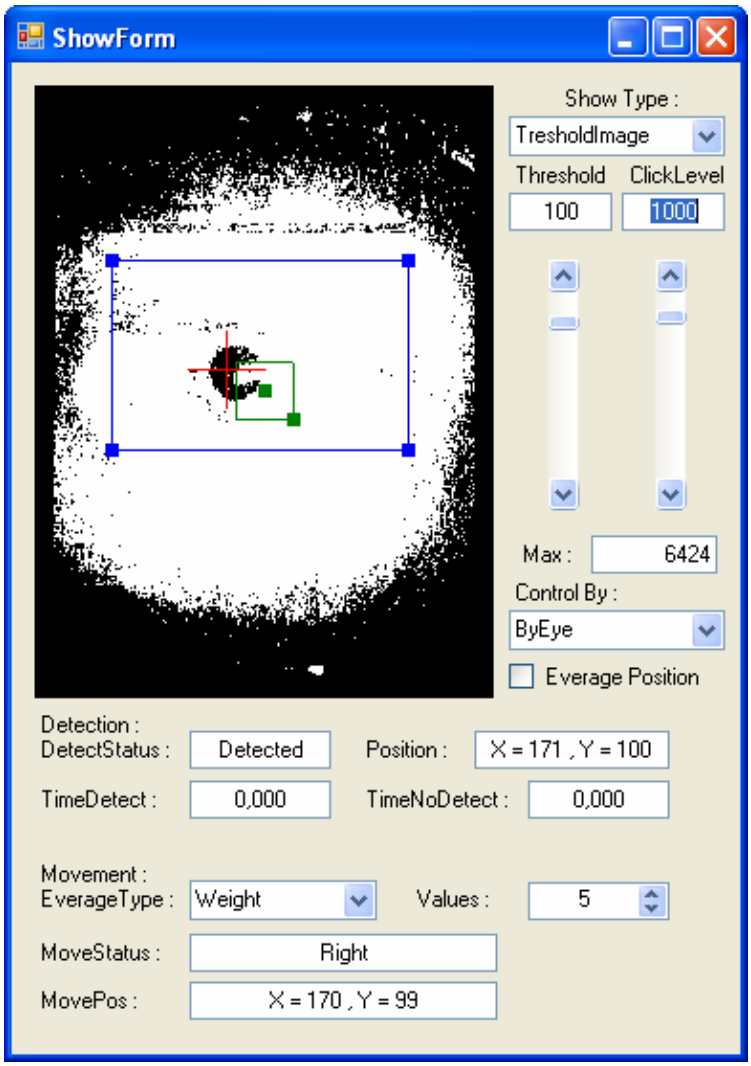


Figure 6.5: Screenshot of the SW interface supporting development of detection algorithms.

Both calibration steps are necessary and they are responsible for a correct function of the I4Control® system for the individual user. This is straightforward for the balanced position, which helps to specify gaze direction. Proper choice of the appropriate detection zone impacts the efficiency of the image processing algorithm as it specifies the size of the processed image and consequently it influences the speed of the processing of the considered image. As it gets smaller the number of the black points decreases and the processor is able to finish processing of more images and consequently it can provide more accurate results.

But it should not be too small: all places where the pupil can appear have to be covered. Results of calibration can be saved for re-use during the next session of the same user.

Further improvements of the I4Control® system have to be achieved through refinement of its image evaluation algorithm. To simplify testing of its various new modifications we have designed and implemented a SW interface which makes it possible to set manually individual parameters of the detection algorithm (e.g. the minimal number of dark point in the close neighbourhood, size of the neighbourhood, etc.). In this way we can study influence of various types of setting (illumination, colour conditions, anatomical properties of the user) on the function of the I4Control® system and to use the gained experience for its further improvement. This interface proved very helpful and the obtained experimental results are rather encouraging.

To support further development of more efficient algorithms for eye image evaluation we have decided to build a database consisting of classified sequences of eye images of various persons looking to different directions. This collection will follow the recommendations provided by Deliverable D2.2 [56] and it will be used for testing new versions of evaluation algorithms.

6.4 Silicon Retina

COGAIN partner UNIZH (Universität Zürich), represented by Tobi Delbruck and Patrick Lichtsteiner, visited partner DTU (Danmarks Tekniske Universitet) represented by Bjarne Ersbøll at DTU (Lyngby, Denmark) and presented two invited talks on their asynchronous dynamic silicon retina at the VisionDay meeting (http://www.visionday.dk/). Subsequently, inspired by discussions with Ersbøll and with partner ITU (IT University of Copenhagen) represented by John-Paulin Hansen, Delbruck evolved previously developed pupil tracker code. A few ‘frames’ of the resulting tracking are shown below. Each frame is a 20 ms slice of retina activity with pupil tracking superimposed. The pupil tracking is based on a very simplified pupil model as a disk of sensitivity, within which the events from the retina drive the location of the pupil. Events on the outside or inside edges of the disk pull the pupil model in the respective directions. This event-driven model receives little input during eye fixations, so the disk does not move. The disk is thus moved directly by data. This simple model will need to be evolved. Partners UNIZH, DTU, and ITU agreed to apply for a mobility grant for a diploma or masters student to combine the knowledge of retina functionality from UNIZH and probabilistic vision processing from DTU and ITU. Partner UNIZH also agreed to supply data and a retina board to partner DTU when more engineering samples have been constructed.

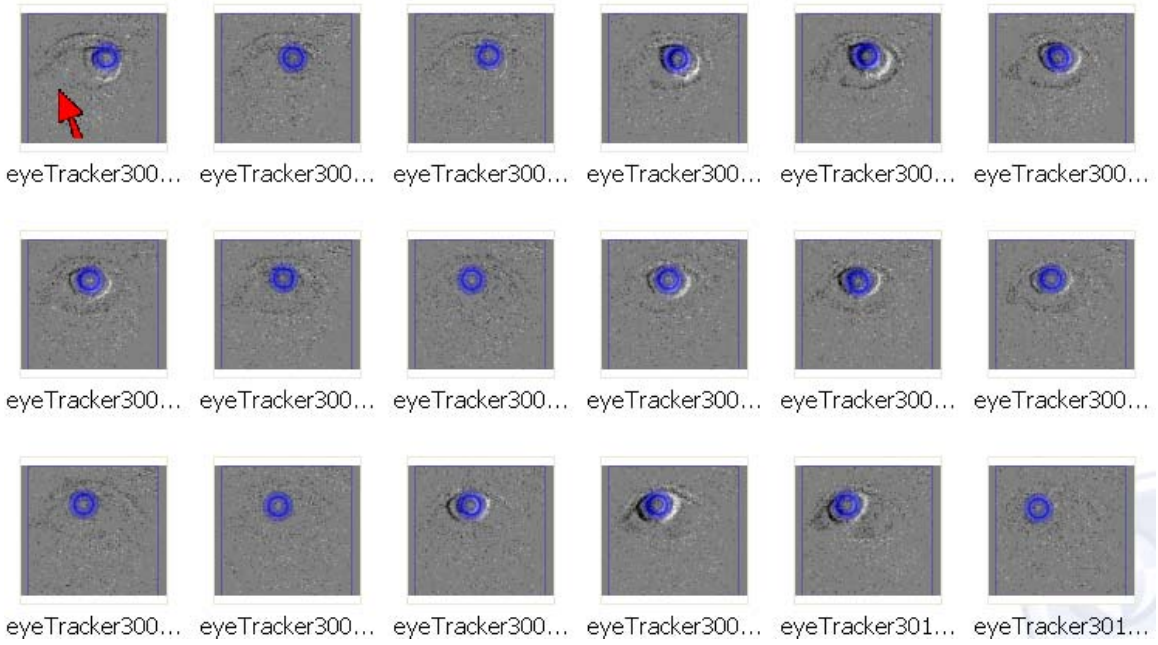


Figure 6.6: A few frames of pupil tracking using the event-based output from the silicon retina.

7 References

1. Young, L.R., & Sheena, D. Survey of eye movement recording methods. *Behavior Research Methods and Instrumentation*, 7(5), (1975) 397-429.
2. Morimoto, C.H., & Mimica, M.R.M.: Eye gaze tracking techniques for interactive applications. *Computer Vision and Image Understanding* 98, (2005) 4–24
3. Yoo, D.H., & Chung, M.J.: A novel non-instrusive eye gaze estimation using crossratio under large head motion. *Computer Vision and Image Understanding* 98 (2005) 25–51
4. Shih, S.W., Wu, Y.T., & Liu, J.: A calibration-free gaze tracking technique. In: *Proceedings of the 15th International Conference on Pattern Recognition*. (2000) 201–204
5. Ohno, T., & Mukawa, N.: A free-head, simple calibration, gaze tracking system that enables gaze-based interaction. In: *Eye Tracking Research and Applications (ETRA)*. (2004) 115–122
6. Brolly, X.L.C., & Mulligan, J.B.: Implicit calibration of a remote gaze tracker. In: *Proceedings of the 2004 Conference on Computer Vision and Pattern Recognition Workshop (CVPRW '04)*. Volume 8. (2004) 134
7. Beymer, D., & Flickner, M.: Eye gaze tracking using an active stereo head. In: *Proceedings of Computer Vision and Pattern Recognition (CVPR)*. Volume 2. (2003) 451–458
8. Tobii: Tobii 1750 eye tracker, Tobii Technology AB, Stockholm, Sweden) http://www.tobii.se.
9. Mulligan, J. Optical eye models for gaze tracking. *ACM Eye Tracking Research and Applications, ETRA 2006*, ACM Press, 51-51.
10. Guestrin, D., & Eizenman, M. General theory of remote gaze estimation using the pupil center and corneal reflections. *IEEE Transactions on Biomedical Engineering*, 53(6), 2006, IEEE, 1124-1133.
11. Ohno, T., Mukawa, N., & Yoshikawa, A. Freegaze: a gaze tracking system for everyday gaze interaction. *ACM Eye Tracking Research and Applications, ETRA 2002*, ACM, pp. 125-132.
12. Allison R.S., Eizenman M., & Cheung B.S., Combined head and eye tracking system for dynamic testing of the vestibular system, *IEEE Trans. Biomed. Eng*., vol. 43, No. 11, 1996, 1073-1082.
13. Fejtová, M., Fejt, J., & Štěpánková, O.: Towards Society of Wisdom. In *Interdisciplinary Aspects of Human-Machine Co-existence and Co-operation*. Praha: Vydavatelství ČVUT, 2005, 253-261, (ISBN 80-01-03275-2).
14. SwissRanger SR-3000, CSEM SA, Zurich, Switzerland.
15. Wyatt H. J., The Form of the Human Pupil, *Vision Res.,* vol. 35, No14, pp.2021-2036, 1995.
16. Corbett, M. C., Rosen, E. S., & O'Brart D. P. S. Corneal Topography: Principles and Applications, London: BMJ Books, p.6.
17. Li, D., Winfield, D., & Parkhurst, D.J. Starburst: A hybrid algorithm for video-based eye tracking combining feature-based and model-based approaches. *Proceedings of the 2nd IEEE CVPR Workshop on Vision for Human-Computer Interaction,* San Diego, USA, 2005.
18. Li, D., Babcock, J., & Parkhurst, D.J. openEyes: A Low-Cost Head-Mounted Eye-Tracking Solution. *ACM Eye Tracking Research & Applications Symposium*, San Diego, USA, pp. 95-100, 2006.
19. Li, D. & Parkhurst, D.J. openEyes: An open-hardware open-source system for low-cost eye tracking. *Journal of Modern Optics*, 53(9), pp. 1295-1311, 2006.
20. Hansen, D.W., & Pece, A.E.C. Eye Tracking in the wild. *Computer Vision and Image Understanding* 98 (2005) 155–181.

21. Zhu, Z., & Ji, Q. Robust real-time eye detection and tracking under variable lighting conditions and various face orientations . *Computer Vision and Image Understanding* 98 (2005) 124–154.
22. Yuille, A.L., Hallinan, P.W., & Cohen, D. Feature extraction from faces using deformable templates, Int.
23. Edwards, G., Cootes, T.F., & Taylor, C.J. Face recognition using active appearance models, in: *ECCV_98. 5th European Conf. on Computer Vision. Proc*, vol. 2, Springer-Verlag, 1998, pp. 581–95.
24. Cootes, T.F., & Taylor, C.J. Active shape models—smart snakes. *Proc. British Machine Vision Conf., BMVC92*, 1992, pp. 266–275.
25. Hansen, D.W., Hansen, J.P., Nielsen, M., Johansen, A.S., & Stegmann, M.B. Eye typing using markov and active appearance models. *IEEE Workshop on Applications on Computer Vision*, 2003, 132–136.
26. Kawato, S., & Tetsutani, N. Detection and tracking of eyes for gaze-camera control, *VI02*, 2002, p. 348.
27. Yang, J., Stiefelhagen, R., Meier, U., & Waibel, A. Robust detection of facial features by generalized symmetry, in: Int. *Conf. on Pattern Recognition*. vol. I, 1992, pp. 117–120.
28. Herpers, R., Michaelis, M., Lichtenauer, K., & Sommer, G. Edge and keypoint detection in facial regions, in: Int. *Conf. on Automatic Face and Gesture Recognition*, 1996.
29. Nixon, M. Eye spacing measurements for facial recognition, *Appl. Digital Image Process*. 575 (VIII) (1985) 279–285.
30. Young, D., Tunley, H., & Samuels, R. *Specialised hough transform and active contour methods for realtime eye tracking.* Tech. Rep. 386, School of Cognitive and Computing Sciences, University of Sussex, 1995.
31. Loy, G., & Zelinsky, A. Fast radial symmetry for detecting points of interest, *PAMI* (2003) 959–973.
32. Viola, P., & Jones, M. Robust real-time face detection, in: Int. *Conf. on Computer Vision*, vol. II, 2001, p. 747.
33. Hansen. D. W., & Pece, A.E.C. Iris tracking with feature free contours. In *Proc. workshop on Analysis and Modelling of Faces and Gestures: AMFG 2003,* October 2003.
34. Carstensen, J.M. *Image analysis, vision and computer graphics.* Technical University of Denmark, Kgs. Lyngby, 2 edition, 2002.
35. Cootes, T. *Statistical models of appearance for computer vision.* A technical report available from http://www.isbe.man.ac.uk/~bim/Models/app_models.pdf, 2004.
36. Baker, S., Matthews, I. & Schneider, J. Automatic construction of active appearance models as an image coding problem. *IEEE Transactions on Pattern Analysis and Machine Intelligen*ce, 26(10), October 2004.
37. Stegmann, M.B., Ersbøll, B.K. & Larsen, R. FAME – a flexible appearance modelling environment. *IEEE Trans. on Medical Imaging*, 22(10):1319–1331, 2003.
38. Tingleff, O., Madsen, K. & Nielsen, H.B. Methods for Non-linear Least Squares Problems. *Lecture Note in Computer Science 02611 Optimization and Data Fitting*, 2004.
39. Qiang Ji. 3D Face pose estimation and tracking from a monocular camera // *Image Vision Computing*. – 2002. – No. 20. – P. 499 – 511.
40. Rae R., & Ritter H.J. Recognition of human head orientation based on artificial neural networks // IEEE *Transactions on Neural Networks*. – 1998. – No. 9. – P. 257 – 265.
41. Darrell T., Moghaddam B., & Pentland A.P. Active face tracking and pose estimation in an interactive room // *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. – 1996.
42. Gee A., Cipolla R. Fast visual tracking by temporal consensus // *Image and Vision Computing*. – 1996. – No. 14. – P. 105 – 114.
43. Horprasert A.T., Yacoob Y., & Davis L.S. Computing 3D head orientation from a monocular image sequence // *Proceedings of SPIE 25th, AIPR Workshop: Emerging Applications of Computer Vision*. – 1996. – No. 2962. – P. 244 – 252.

44. Dervinis, D. *Extracting characteristic face points from 2D image* // Electronics and Electrical Engineering. – Kaunas: Technologija, 2004.– No.5(54). – P. 16 – 20. (in Lithuanian).
45. Yang J., & Waibel A. A real-time face tracker // In *Proceedings of the Third IEEE Workshop on Applications of Computer Vision*. – Sarasota, FL,1996. – P142–147.
46. Open source computer vision library. Web link: http://www.intel.com/technology/computing/opencv/
47. 47 Qingcang Yu, & Harry H. Interactive Open Architecture Computer Vision//*ICTAI'03, 15th IEEE International Conference on Tools with Artificial Intelligence*. – 2003. – P. 406.
48. The Color FERET Database. Web link: http://www.nist.gov/humanid/colorferet/colorferet.html.
49. Cohn–Kanade, AU–Coded Facial Expression Database. Web link: http://www.cs.cmu.edu/~face
50. Meyer, A., Böhme, M., Martinetz, T. & Barth, E. A single-camera remote eye tracker. In Perception and Interactive Technologies, volume 4021 of Lecture Notes in Artificial Intelligence, pages 208-211. Springer, 2006.
51. Böhme, M., Meyer, A., Martinetz, T. & Barth, E. Remote Eye Tracking: State of the Art and Directions for Future Development. To appear in: *2nd Conference on Communication by Gaze Interaction - COGAIN 2006*.
52. Fejtová, M., Fejt, J. & Lhotská, L.: Controlling a PC by Eye Movements: The MEMREC Project. In *Computers Helping People with Special Needs*. Berlin: Springer, 2004, s. 770-773. ISBN 3-540-22334-7.
53. Fejtová, M., & Fejt, J.: System I4Control: The Eye As a New Computer Periphery. In *The 3rd European Medical and Biological Engineering Conference - EMBEC´05* [CD-ROM]. Praha: Společnost biomedicínského inženýrství a lékařské informatiky ČLS JEP, 2005, vol. 11, ISSN 1727-1983.
54. Fejtová, M., Fejt, J., Novák, P., & Štěpánková, O.: System I4Control®: Contactless control PC. In *Proceedings of IEEE 10th In Conference on Intelligent Engineering Systems 2006* [CD-ROM]. Los Alamitos: IEEE Computer Society, 2006, ISBN 1-4244-9709-6.
55. Fejtová, M., Fejt, J., & Štěpánková, O.: Eye as an Actuator. In *Computers Helping People with Special Needs*. Berlin: Springer, 2006, vol. 4061, s. 954-961. ISBN 3-540-36020-4.
56. Bates, R., Istance, H., & Spakov, O. (2006) *D2.2 Requirements for the Common Format of Eye Movement Data***.** Communication by Gaze Interaction (COGAIN), IST-2003-511598: Deliverable 2.2.
57. Vester-Christensen, M., Leimberg, D., Ersbøll, B.K., & Hansen, L.K.. Towards emotion modeling based on gaze dynamics in generic interfaces. *Proceedings of HCI International 2005.* Proceedings are available on CD-ROM by Lawrence Erlbaum Associates, Inc (ISBN 0-8058-5807-5).