Available online at www.sciencedirect.com

SCIENCE DIRECT®

Computer Vision and Image Understanding 98 (2005) 155–181

Computer Vision and Image Understanding

www.elsevier.com/locate/cviu

Eye tracking in the wild

Dan Witzner Hansen[a,*], Arthur E.C. Pece[b,c]

[a] *IT University Copenhagen, Rued Langgaardsvej 7, 2300 Copenhagen S, Denmark*
[b] *Heimdall Vision, Bjørnsonsvej 29, 2500 Valby, Denmark*
[c] *Department of Computer Science, University of Copenhagen, Universitetsparken 1, 2100 Copenhagen, Denmark*

Received 27 July 2004; accepted 27 July 2004
Available online 6 October 2004

Abstract

An active contour tracker is presented which can be used for gaze-based interaction with off-the-shelf components. The underlying contour model is based on image statistics and avoids explicit feature detection. The tracker combines particle filtering with the EM algorithm. The method exhibits robustness to light changes and camera defocusing; consequently, the model is well suited for use in systems using off-the-shelf hardware, but may equally well be used in controlled environments, such as in IR-based settings. The method is even capable of handling sudden changes between IR and non-IR light conditions, without changing parameters. For the purpose of determining where the user is looking, calibration is usually needed. The number of calibration points used in different methods varies from a few to several thousands, depending on the prior knowledge used on the setup and equipment. We examine basic properties of gaze determination when the geometry of the camera, screen, and user is unknown. In particular we present a lower bound on the number of calibration points needed for gaze determination on planar objects, and we examine degenerate configurations. Based on this lower bound we apply a simple calibration procedure, to facilitate gaze estimation.

Keywords: Eye tracking; Gaze estimation; Contour tracking; Expectation maximization; Particle filter; Condensation

[*] Corresponding author.
E-mail addresses: witzner@itu.dk (D.W. Hansen), aecp@heimdall-vision.com (A.E.C. Pece).

1077-3142/$ - see front matter © 2004 Elsevier Inc. All rights reserved.
doi:10.1016/j.cviu.2004.07.013

1. Introduction

Humans acquire a vast amount of information through the eyes, and the eyes in turn reveal information about our attention and intention. Detection of the eye gaze enables collection of valuable information for use in psychophysics and human–computer interaction (HCI). The use of commercial off-the-shelf (COTS) products as elements in larger systems is becoming increasingly commonplace. Using COTS for camera-based eye tracking tasks has many advantages, but it certainly introduces several new problems as less assumptions on the system can be made. Eye tracking based on COTS holds potential for a large number of possible applications such as in the entertainment industry and for eye typing [23,46,9].

If cost is a constraint, it is usually not possible to exploit IR light sources and other specialized devices as they cannot be bought in a common hardware store. On the same token pan-and-tilt cameras cannot be used, thus forcing such systems to be passive. Very little control over the cameras and the geometry of the setup can be expected. The methods employed for eye tracking should therefore be able to handle changes in light conditions and image defocusing, and through view and scale changes. Users may buy cameras with built-in IR light emitters. It is therefore desirable that the method is applicable to images obtained both through IR and those that do not, without changing the model or the parameters used in the model. By avoiding feature detection, the proposed method becomes robust towards changes in illumination and image defocusing.

Designing systems for the general public, it is unrealistic to assume that people are able to do camera calibration and make accurate setups of the camera, monitor, and user correctly. It is not obvious how the relaxed assumptions on the setup influence the number of calibrations points needed for gaze estimation. Obviously the more calibration points used, the better the chances are to be able to infer the mapping from the image to the monitor. It would even be possible to sample the entire function space given sufficiently many calibration points. From the point of view of the user, a low number of calibration points is preferred as calibration may be perceived as a tedious procedure. Systems that require many calibration points for every session are therefore not likely to succeed.

In this paper, we propose a method for iris tracking which does not require explicit feature detection. The method is shown to be able to track the iris under changing light conditions and image blurring without setting thresholds. Second we propose a lower bound on the number of calibration points needed for screen-based applications and in settings where the geometry between the user, monitor and camera is unknown. The main contributions of this paper are: (1) the application of the Marginalized Contour Model for iris tracking; (2) the combination of particle filtering (to keep track of the iris) and Expectation Maximization (to obtain accurate pose estimation); and (3) a lower bound on the number of calibration points needed for gaze estimation.

The rest of the paper is organized as follows. Section 2 reviews related work and Section 3 describes the overall method. Section 4 derives the marginalized contour model and Section 5 describes the EM contour algorithm. The results of iris tracking and blink detection are given in Section 6. Section 7 derives a lower bound on the number of calibration points needed in scenarios where the geometry is fixed and shows the results of using the lower bound for gaze estimation.

2. Eye tracking

Detection of the human eye is in general a difficult task as the contrast between eye and skin is generally poor. In addition, on all sides of the socket rim, the eye is protected by structures of bone: the high nasal one to the side; the overhanging brow ridge; and to the outside the protruding cheek mound. These structures protect the eyes, but may result in poor illumination for eye tracking. The appearance of the pupil is heavily influenced by occlusions from the eye lids and may often be totally covered. The effects of occlusion and illumination changes is also related to the ethnic origin of the user. Both Caucasian and Asian people have facial features that may make iris tracking difficult. The eye lids of Asians are generally close together and may thus result in less boundary information. On the other hand, the nasal bone and superciliary arch of Caucasians are usually more pronounced and therefore casts more shadows on the eye. Once the eye is located, the iris is a prominent and reliable feature within the eye region because of its high contrast.

2.1. Related work

Methods used for eye detection and tracking rely on different prior assumptions on the image data and in general two classes of approaches exist. One common and effective approach is to exploit active illumination from infrared (IR) light emitters. Through novel synchronization schemes and by using the reflective properties of the pupil when exposed to near infrared light (dark or bright pupil effects) the eye can be detected and tracked effectively. In addition to controlling the light conditions, IR also plays an important role for some gaze estimation methods. Other approaches avoid the use of active illumination, but rely solely on natural light. This make the problem of detection much harder as less assumptions on the image data can be made. Methods based on active light is the most predominate in both research [5,52,27,41,53] and in commercial systems [40,37,20]. Ebisawa and Satoh [5] use a novel synchronization scheme in which the difference between images obtained from on axis and off axis light emitters are used for tracking. Kalman filtering [15], the Mean shift algorithm [52], and combinations of Kalman and mean shift filtering [53] are applied for eye tracking. The success of these approaches 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. Methods using IR can therefore be less reliable in these situations and several methods exist that address these issues [5,52]. IR light and synchronization schemes can in general not be exploited when using COTS for eye tracking as IR light emitters cannot be bought off-the-shelf. These schemes will consequently not be considered explicitly in the proposed method. However, the method in this paper is, without changing the model, also capable of tracking the iris in IR light.

Eye tracking and detection methods fall broadly within three categories, namely deformable templates, appearance-based, and feature-based methods. 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. Grauman et al. [7] uses background subtraction and anthropomorphic constraints to initialize a correlation-based tracker. Matsumoto and Zelinsky [25] and Newman et al. [28] present trackers based on template matching and stereo cameras. Excellent tracking performance is reported, but the method requires a fully calibrated stereo setup and a full facial model for each user.

The appearance of eye regions share commonalities across race, illumination and viewing angle. Rather than relying on a single instance of the eye region, the eye model can be constructed from a large set of training examples with varying pose and light conditions. Based on the statistics of the training set a classifier can be constructed for detection purposes over a larger set of subjects. Neural networks [16,33] and Support Vector Machines [52] are methods that employ this strategy. Eye region localization by Eigenimages [26] uses a subset of the principal components of the training data to construct a low-dimensional object subspace to represent the image data. Recognition is performed by measuring distances to the object subspace. The limitations of methods purely based on detection of eyes in individual frames is that they do not make use of prior information from previous frames. Even a 99% accurate detection system will fail every four seconds. This can be avoided by temporal filtering. They may on the other hand be more useful for initialization and validation of hypotheses.

Deformable template-based method [51,6,2] rely on a generic template which is matched to the image. In particular deformable templates [51], 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. Hallinan [8] uses statistical measures in a deformable template approach to account for statistical variations. The method uses an idealized eye consisting of two regions with uniform intensity. One region corresponds to the iris region and the other the area of the sclera. Ivins and Porrill [14] describe a method of tracking the three-dimensional motion of the iris in a video sequence. A five-parameter scalable and deformable model is developed to relate translations, rotation, scaling due to changes in eye-camera distance, and partial scaling due to expansion and contraction of the pupil. The method requires very high-quality and high-resolution images. Lam and Yan [19] propose a method using snakes to determine the outline of the head. The approximate positions of the eyes are then found by anthropomorphic averages. Detected eye corners are used to reduce the number of iterations of the optimization of a deformable template. This model consists of parabolas for the eyelids and a subset of a circle for iris outline. A speedup is obtained compared to Yuille et al. [51] by exploiting the positions of the corners of the eye. This method requires the presence of four corners on the eye, which, in turn, only occur if the iris is partially occluded by the upper eyelid. When the eyes are wide open, the method fails as these corners do not exist. Deng and Lai [4] use a combination of deformable template and edge detection. An extended iris mask is used to select edges of iris obtained through an edge image. The template is initialized by manually locating the eye region. Its parameters are also similarly initialized. Once this is done the template is allowed to deform in an energy minimization manner. The position of the template in an initial frame is used as a starting point for deformations that are carried out in successive frames. The faces must be nearly frontal-view and the image of the eyes should be large enough to be described by the template.

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. [9] 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 [18] and Yang et al. [48] 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. [10] 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. [49] 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 [21] proposes the Fast Radial Symmetry Transform 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 [43]. 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.

3. Method overview

The proposed method is based on recursive estimation of the state variables of the iris. Given a sequence of T frames, at time t only data from the previous $t-1$ images are available. The states and measurements are represented by $\mathbf{x}_t$ and $\mathbf{y}_t$, respectively, and the previous states and measurements are represented $\underline{\mathbf{x}}_t = (\mathbf{x}_1, \ldots, \mathbf{x}_t)$ and $\underline{\mathbf{y}}_t = (\mathbf{y}_1, \ldots, \mathbf{y}_t)$. At time t the observation $\mathbf{y}_t$ is assumed independent of the previous state $\mathbf{x}_{t-1}$ and previous observation $\mathbf{y}_{t-1}$ given the current state $\mathbf{x}_t$.

Employing these assumptions the tracking problem can be stated as a Bayesian inference problem in the well known recursive relation [13]:

$$p(\mathbf{x}_{t+1}|\underline{\mathbf{y}}_{t+1}) \propto p(\mathbf{y}_t|\mathbf{x}_t)p(\mathbf{x}_{t+1}|\underline{\mathbf{y}}_t), \tag{1}$$

$$p(\mathbf{x}_{t+1}|\underline{\mathbf{y}}_t) = \int p(\mathbf{x}_{t+1}|\mathbf{x}_t)p(\mathbf{x}_t|\underline{\mathbf{y}}_t)\,\mathrm{d}\mathbf{x}_t. \tag{2}$$

The method combines particle filtering with the Expectation Maximization (EM) algorithm. Particle filtering is used, as it allows to maintain multiple hypotheses which make it robust in clutter and capable of recovering from occlusion. Particle filtering is particularly suitable for iris tracking, because changes in iris position are fast and do not follow a smooth and predictable pattern. Particle filters generally require a large set of particle to accurately determine the pose parameters. By contrast, the method uses a fairly small set of particle to maintain track of the object while using the EM Contour method for precise pose estimation. In this way computation time is lowered while maintaining accuracy. The EM Contour algorithm is described in Section 5.

The aim of particle filtering is to approximate the filtering distribution $p(\mathbf{x}_t|\underline{\mathbf{y}}_t)$ by a weighted sample set $\mathbf{S}_t^N = \{(\mathbf{x}_t^{(n)}, \pi_t^{(n)})\}_{n=1}^N$, where $\mathbf{x}_t^{(n)}$ is the nth instance of a state at time t with weight $\pi_t^{(n)}$. This sample set will evolve into a new sample set $\mathbf{S}_{t+1}^N$, representing the posterior pdf (probability density function) $p(\mathbf{x}_{t+1}|\underline{\mathbf{y}}_{t+1})$ at time $t+1$. The object location in the particle filter is usually represented by the sample mean. *Factored sampling* is utilized in the CONDENSATION approach to particle filtering ([13]): the samples are drawn from the prediction prior $p(\mathbf{x}_{t+1}|\underline{\mathbf{y}}_t)$, and sample weights are proportional to the observation likelihood $p(\mathbf{y}_t|\mathbf{x}_t)$. This approach is employed here.

The robustness of particle filters lies in maintaining a set of hypothesis. Generally, the larger the number of hypotheses, the better the chances to get accurate tracking results, but the slower the tracking speed. Therefore, there is a trade-off between tracking accuracy and speed.

Using particle filters in large images may require a large set of particles to sufficiently sample the spatial parameters. Adding samples to the particle set may only improve accuracy slowly, due to the sampling strategy employed (see Section 6). This added accuracy may become costly in terms of computation time. To lower the requirements on the number of particles while improving tracking performance, we propose to use an image scale space H_I^M, with M image scales. Particle filtering is performed at the coarsest scale $H_I^{(0)}$. The EM Contour algorithm is applied to gradually finer image scales $H_I^{(i)}$ $(0 \leqslant i < M)$ using the estimate from each scale for initialization at the next finer scale, and the sample mean from the particle filter for initialization at the coarsest scale. In this way the particle filter samples the posterior more effectively, while the EM Contour algorithm reaches the (local) maximum-likelihood estimate of the iris location. Fig. 1 illustrates the flow diagram of the method, and Fig. 2 describes the algorithm for a single frame of an image sequence.

3.1. State model and dynamics

The model underlying the tracker is given in Fig. 3. The model consists of three components:

a dynamical model defining the pdf over the iris state at the current image frame, given the state in the previous frame;

a geometric model defining a pdf over contours on the image plane, given the iris state at the current frame;

an observation model defining a pdf over gray-level differences, given the contour

Section 4 describes the observation model; the geometric and dynamical models are described in the following section.

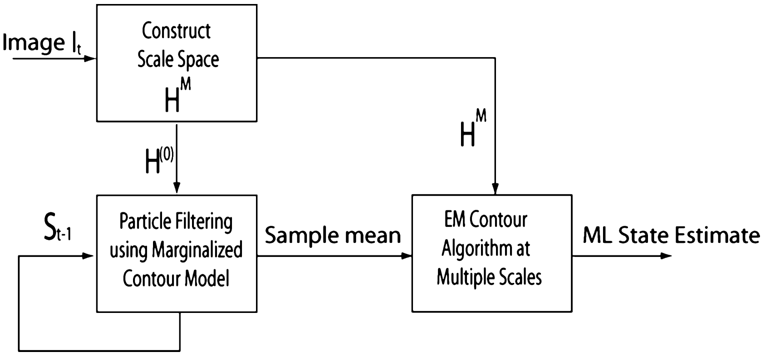
Image I_t
Construct Scale Space H^M
$H^{(0)}$
H^M
S_{t-1}
Particle Filtering using Marginalized Contour Model
Sample mean
EM Contour Algorithm at Multiple Scales
ML State Estimate

Fig. 1. Overview of each iteration in the algorithm. Overall tracking is performed on the coarsest image scale through particle filtering; starting from the weighted mean of the particle states (sample mean), maximum-likelihood estimation of the object state is performed through the EM contour algorithm over gradually finer image scales.

```
Contour Tracker: operation for a single image frame
Input: Image scale space $H_I^M$ of $I$, Motion model $\Sigma$ and particle set $S_{t-1}$
Output: Optimized mean state $\bar{\mathbf{x}}$ and new particle set $S_t$

  (1) Obtain the sample set $S_t$ by selecting $N$ samples proportionally to their weight
      from sample set $S_{t-1}$
  (2) Predict all samples in $S_t$ according to equation (3)
  (3) Update weights in $S_t$ according to equation (17) using $H_I^{(0)}$ as reference image.
  (4) Calculate sample mean
          $\bar{\mathbf{x}}_0 = \frac{\sum_{i=1}^{N} \pi_i \mathbf{x}^{(i)}}{\sum_{i=1}^{N} \pi_i}$
  (5) $\tilde{\mathbf{x}}_\mathbf{0} = \mathscr{O}_0(\bar{\mathbf{x}}_0)$ where $\mathscr{O}_i(\mathbf{x}*)$ represents the EM Contour algorithm applied at
      scale $i$ with initialization given by the argument $\mathbf{x}*$.
  (6) for $i = 1 : M\ \tilde{\mathbf{x}}_\mathbf{i} = \mathscr{O}_i(\tilde{\mathbf{x}}_{i-1})$
  (7) $\bar{\mathbf{x}} = \tilde{\mathbf{x}}_M$
```

Fig. 2. The Iris Tracker as applied to a single frame of an image sequence.

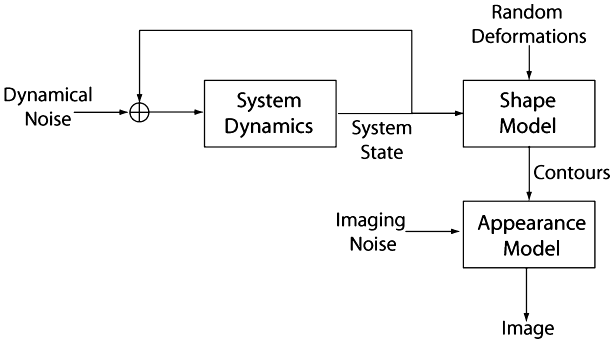

Fig. 3. Generative model underlying the tracker: the object state evolves over time through the dynamical model and generates a contour on the image plane.

The iris appears elliptical on the image plane; therefore we model it as an ellipse and the state $\mathbf{x}$ is given by five state variables:

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

where $(c_x,\ c_y)$ is the center of the iris, λ_1, λ_2 are the major and minor axes, and θ the angle of the major axis with respect to the vertical. These are the variables being estimated in the method.

Pupil movements can be very rapid from one image frame to another. The dynamics is therefore modelled as a first order auto regressive process using a Gaussian noise model:

$$\mathbf{x}_{t+1} = \mathbf{x}_t + \mathbf{v}_t, \quad \mathbf{v}_t \sim \mathscr{N}(0, \Sigma_t), \tag{3}$$

where Σ_t is the time dependent covariance matrix of the noise $\mathbf{v}_t$. The time dependency is included to compensate for scale changes: when the apparent size of the eye increases, the corresponding eye movements can also be expected to increase. For this reason, the first 2 diagonal elements of Σ (corresponding to the state variables c_x and c_y) are assumed to be linearly dependent on the previous sample mean.

4. Observation model

This section describes the observation model that defines the pdf $p(\mathbf{y}_t|\mathbf{x}_t)$. The model was introduced by Pece and Worrall [31]. Most active contour methods can be divided into two main classes, based on the method for evaluating the image evidence. One class relies on assuming object edges generate image features and thus depends on the extraction of features from the image [13,39]. The pose and shape of contours are estimated by minimizing the squared distances between contours and image features. The assumption behind the feature-based methods is that edges generate image features. From a statistical point of view this approach throws away information and from a practical point of view it is difficult to set appropriate thresholds that applies for large changes in image quality. Apart from the problem of finding correct correspondences, the thresholds necessary for feature detection inevitably make these methods sensitive to noise. Other active-contour methods [17,31,50] avoid feature detection by maximizing feature values (without thresholding) underlying the contour, rather than minimizing the distance between locally strongest feature and contour. In this way there is no information loss. The underlying idea is that a large image gradient is likely to arise from a boundary between object and background. The method introduced here is of the latter class, but smoothing as in [50] is replaced by marginalization over possible deformations of the object shape.

4.1. Assumptions

The model is based on the following assumptions:

1. The pdf of the observation depends only on the gray level differences (GLD's).
2. Gray level differences between pixels along a line are statistically independent.
3. Intensity values of nearby pixels are correlated if both belong to the object being tracked or both belong to the background: thus, a priori statistical dependencies between nearby pixels is assumed.
4. There is no correlation between pixel values if they are on opposite sides of the object boundary
5. The shape of the contour is subject to random local variability, which means that marginalization over local deformations is required for a Bayesian estimate of the contour parameters.

Similar assumptions can be found separately in the literature, e.g., [22] for the last assumption. Taking the assumptions together means that no features need to be detected and matched to the model (leading to greater robustness against noise), while at the same time local shape variations are taken explicitly into account. As shown below, this model leads to a simple closed-form expression for the likelihood of the image given the contour parameters [31].

4.2. Definitions

Define a normal to a given point on the contour as the *measurement line*. Define the coordinate v, on the measurement line. Given the position μ of the contour on the measurement line, the distance from μ to a point v is $\varepsilon = v - \mu$. $\eta(v)$ is a binary indicator variable which is 1 if the boundary of the target is in the interval $[v - \Delta v/2, v + \Delta v/2]$ (with regular inter-point spacing Δv) on the measurement line; and 0 otherwise. Denote the gray level difference between two points on the measurement line by $\Delta I(v) \equiv I(v + \Delta v/2) - I(v - \Delta\ v/2)$, and the *observation* on a given measurement line by $\mathbf{I} = \{I(i\Delta v)|i \in \mathbb{Z}\}$. These definitions are illustrated in Fig. 4.

4.3. Likelihood of the image

The observations along the measurement line depend on the contour locations in the image. This means that the likelihoods computed for different locations are not comparable, as they are likelihoods of different observations. A better evaluation function is given by the likelihood of the entire image $\mathscr{I}$ given a contour at location μ, as a function $f^*(\mathscr{I}|\mu)$ of the contour location μ (for simplicity of notation we do not include the coordinates of the measurement line on which the contour is located).

Denote by $f_a(\mathscr{I})$ the likelihood of the image given no contour and by $f_R(\mathscr{I}|\mu)$ the ratio $f^*(\mathscr{I}|\mu)/f_a(\mathscr{I})$, then the log-likelihood of the entire image can be decomposed as follows:

$$\log f^*(\mathscr{I}|\mu) = \log f_a(\mathscr{I}) + \log f_R(\mathscr{I}|\mu). \tag{4}$$

The first term on the right-hand side of Eq. (4) involves complex statistical dependencies between pixels, and is expensive to calculate as all image pixels must be inspected. Most importantly, the estimation of this term is needless as it is an additive term which is independent on the presence and location of the contour. Consequently, in order to fit contours to the image, we consider only the log-likelihood ratio $\log f_R(\mathscr{I}|\mu)$. This derivation is fairly standard in the field of active contours, see e.g., [3] and [36].

Note that $f_R(\mathscr{I}|\mu)$ is the ratio between the likelihood for the hypothesis that the target is present (from Eq. (13)); and the null hypothesis that the contour is not present Eq. (7). Hence the likelihood ratio can also be used for testing the hypothesis of the presence of a contour.

4.4. Image statistics

The pdf of gray-level differences between neighboring pixels is well approximated by a generalized Laplacian [11]:

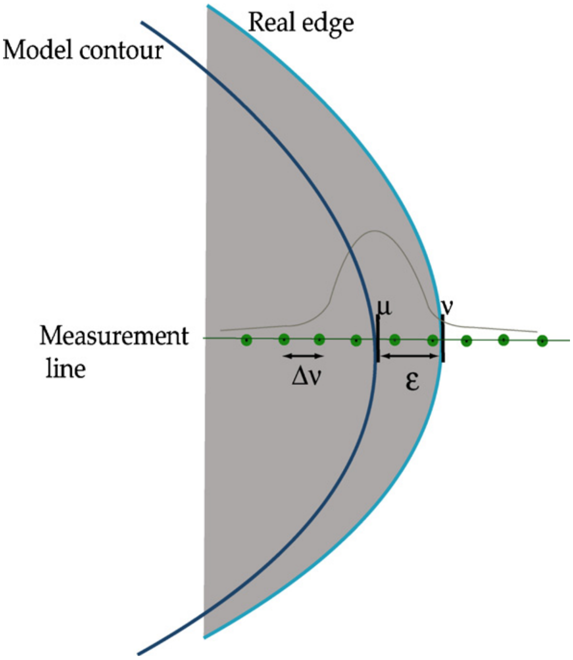

Fig. 4. Marginalized contour definitions

$$f_L(\Delta I) = \frac{1}{Z_L} \exp\left(-\left|\frac{\Delta I}{\lambda}\right|^{\beta}\right), \tag{5}$$

where ΔI is the gray level difference, λ depends on the distance between the two sampled image locations, β is a parameter approximately equal to 0.5 and Z_L is a normalization constant. In the following we assume that $\beta = 0.5$, which implies $Z_L = 4\lambda$.

In Fig. 5 the distribution of gray level differences over images of eyes is shown. The gray level differences, calculated over different scales of Δv, are shown in the figure. It can be seen that the distribution of gray level differences for the class of eye images is well approximated by the generalized Laplacian (defined in Eq. (5)). Further, it can be seen that the width λ of the distribution increases with the scale. This is caused by decorrelation of pixel values as Δv increases.

4.5. Distributions on measurement lines

If there is no known object edge between two points $[v - \Delta v/2, v + \Delta v/2]$ on the measurement line, the pdf of the gray levels follows the generalized Laplacian defined in Eq. (5):

$$f[\Delta I(v)|\eta(v) = 0] = f_L[\Delta I(v)]. \tag{6}$$

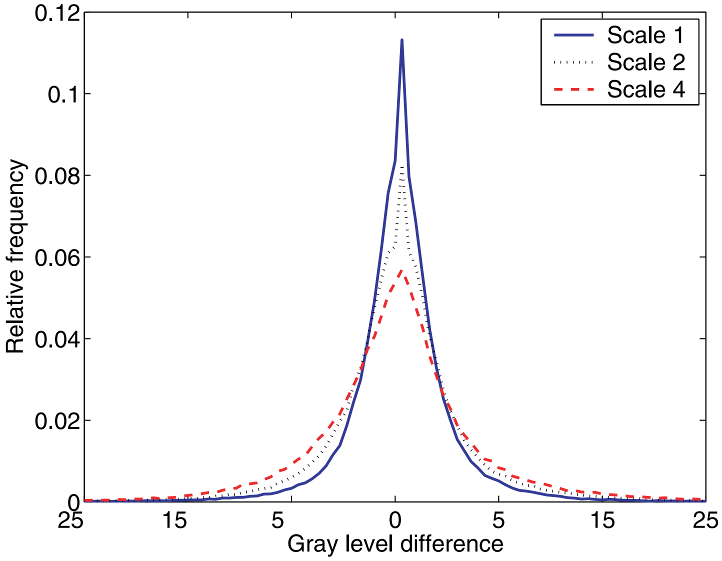

Fig. 5. Distribution of gray level differences over scales for images of eyes. The scale number refers to the distance between gray level measurements: "scale 1" refers to 1 pixel distance and "scale 4" refers to 4 pixels distance between the measurement points.

Therefore, assuming statistical independence between gray level *differences*,[1] the pdf of the observation in the absence of an edge is given by

$$f_a(\mathbf{I}) \equiv \prod_i f_L[\Delta I(i\Delta v)]. \tag{7}$$

Note that the absence of an edge of the object being tracked does not imply the absence of any edge: there can be edges within the background as well as within the object due to unmodelled objects and to surface features.

Two gray levels observed on opposite sides of an edge are assumed to be statistically independent. The conditional pdf of gray level differences, separated by an edge, can be assumed to be uniform for simplicity:

$$f[\Delta I(v)|\eta(v) = 1] \approx \frac{1}{m}, \tag{8}$$

where m is the number of gray levels. If there is a known object boundary at location $j\Delta v$, then this point on the measurement line will correspond to gray level differences across the boundary, the rest will be gray level differences of either object or background. In this case, the pdf of the observation is given by :

$$f_c(\mathbf{I}|j\Delta v) = \frac{1}{m}\prod_{i\neq j} f_L[\Delta I(i\Delta v)] = \frac{1}{m}\frac{f_a(\mathbf{I})}{f_L(\Delta I(j\Delta v))}. \tag{9}$$

[1] Dependencies between *gray levels* are governed by the generalized Laplacian.

4.6. Integrating over deformations

The position of the idealized contour does not exactly correspond to the position of the object boundary, even if the position of the object is known. For simplicity, we assume a Gaussian distribution of geometric deformations of the object at each sample point. In the following, v will denote the location of the object boundary on the measurement line. As mentioned above, μ is the intersection of the measurement line and the (idealized) contour, and the distance from μ to v is $\varepsilon = v - \mu$. The prior pdf of deformations $f_D(\varepsilon)$ is defined by:

$$f_D(\varepsilon) = \frac{1}{Z_D} \exp\left(\frac{-\varepsilon^2}{2\sigma^2}\right), \tag{10}$$

where $Z_D = \sqrt{2\pi}\sigma$ is a normalization factor. This assumption is similar to the one proposed in [22] where a contour point generates a feature at a random distance from the contour with a Gaussian pdf.

The likelihood of the observation $\mathbf{I}$ given the contour location μ and deformation ε is naturally given by

$$f(\mathbf{I}|\mu,\varepsilon) = f_c(\mathbf{I}|\mu+\varepsilon). \tag{11}$$

The joint likelihood of the observation $\mathbf{I}$ and the deformation ε given the contour is:

$$f(\mathbf{I},\varepsilon|\mu) = f_c(\mathbf{I}|\mu+\varepsilon)f_D(\varepsilon). \tag{12}$$

Marginalizing over possible deformations, the likelihood is given by:

$$f_M(\mathbf{I}|\mu) = \int f_c(\mathbf{I}|\mu+\varepsilon)f_D(\varepsilon)\,\mathrm{d}\varepsilon = \frac{1}{m}f_a(\mathbf{I})\int \frac{f_D(\varepsilon)}{f_L(\Delta I(v))}\,\mathrm{d}\varepsilon. \tag{13}$$

The model is illustrated in Fig. 6

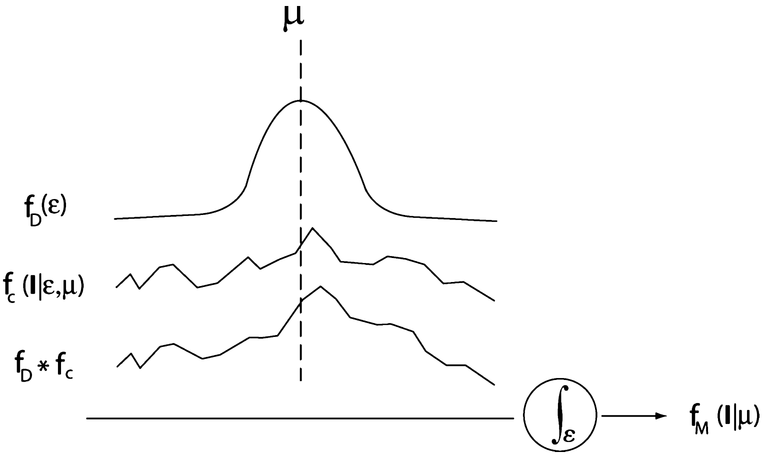

Fig. 6. The likelihood of the contour is obtained by marginalizing the likelihood of an edge over all possible deformations ε.

On the basis of Section 4.3, we are interested in the likelihood ratio given by

$$f_R(\mathbf{I}|\mu) = \frac{f_M(\mathbf{I}|\mu)}{f_a(\mathbf{I})} = \frac{1}{m}\int \frac{f_D(\varepsilon)}{f_L(\Delta I(v))}\,\mathrm{d}\varepsilon. \tag{14}$$

It is convenient to take the logarithm to obtain the log-likelihood ratio:

$$h(\mathbf{I}|\mu) \equiv \log f_R(\mathbf{I}|\mu) = -\log(m) + \log\int \frac{f_D(\varepsilon)}{f_L(\Delta I(v))}\,\mathrm{d}\varepsilon. \tag{15}$$

The integral can be approximated as a finite sum over a discrete set of possible deformations $\varepsilon_j = j\Delta v - \mu$:

$$h(\mathbf{I}|\mu) = -\log(m) + \log\sum_j \frac{f_D(\varepsilon_j)}{f_L(\Delta I(j\Delta v))}\Delta v. \tag{16}$$

This summation is denoted as the *point-evaluation function*. Note that the integral in Eq. (15) is defined from minus to plus infinity. However, due to the Gaussian term, the summation in Eq. (16) only needs to be taken over a finite interval on each side of the contour, e.g., over the interval from -2σ to 2σ.

Using the definitions of the generalized Laplacian, f_L, and the pdf of the deformation, f_D, the point evaluation function above becomes

$$h(\mathbf{I}|\mu) = h_0 + \log\sum_j \exp\left[\sqrt{\frac{|\Delta I(j\Delta v)|}{\lambda}} - \frac{\varepsilon_j^2}{2\sigma^2}\right], \tag{17}$$

where $h_0 = \log Z_l/m - \log Z_\mathrm{D}/\Delta v$.

5. EM Contour algorithm

Expectation Maximization (EM) is an iterative method for maximum-likelihood or maximum a posteriori estimation. It is useful when the underlying model includes intermediate variables between the observed variables (gray levels) and the state variables to be estimated. The EM Contour algorithm is a special case of Expectation Maximization as applied to active contours. A full description is given in [31]. This section gives a short introduction to the algorithm. The way the algorithm is integrated into the iris tracker has been described in Section 3.

For one sample point, the k iteration of the EM contour algorithm consist of the steps

E Estimate the pdf $p^{(k)}(\varepsilon) \equiv f(\varepsilon|\mathbf{I}, \mu^{(k-1)})$ of the deformation given the observation and the estimate of the contour in the previous iteration.

M Maximize the value of the EM functional:

$$\mathscr{F}_p^{(k)}(\mu|\mathbf{I}) \equiv \int_\varepsilon p^{(k)}(\varepsilon)\log f(\mu^{(k-1)}|\mathbf{I},\varepsilon)\,\mathrm{d}\varepsilon. \tag{18}$$

Let p_j denote the probability of the deformation taking a value between $(j-\frac{1}{2})\Delta v$ and $(j+\frac{1}{2})\Delta v$, given the observations and the contour:

$$p_j = \frac{f(f(\varepsilon_j|\mathbf{I},\mu))\Delta v}{\sum_i f(\varepsilon_i|\mathbf{I},\mu)\Delta v} = \frac{f(\mathbf{I},\varepsilon_j|\mu)}{\sum_i f(\mathbf{I},\varepsilon_i|\mu)} = \frac{f_D(\varepsilon_j)f_L^{-1}[\Delta I(j\Delta v)]}{\sum_i f_D(\varepsilon_i)f_L^{-1}[\Delta I(i\Delta v)]}, \tag{19}$$

where Eqs. (12) and (9) were used in the last step.

As in the case of marginalization, Eq. (13), the integral in Eq. (18) will in the following be approximated by a summation. The EM functional can be expanded as follows:

$$\begin{aligned}\mathscr{F}_p^{(k)}(\mu|\mathbf{I}) &= \sum\nolimits_j p_j \log f(\mathbf{I}, j\Delta v|\mu) = \sum\nolimits_j p_j \log f_c(\mathbf{I}|\mu+\varepsilon)f_D(\varepsilon) \\ &= h_0 + \sum p_j\left[\sqrt{\frac{|\Delta\mathbf{I}(j\Delta v)|}{\lambda}} - \frac{\varepsilon_j^2}{2\sigma^2}\right],\end{aligned} \tag{20}$$

Define the center of mass of the observation:

$$\hat{v} \equiv \sum p_j j\Delta v.$$

The definition of the center of mass allows a simplification of Eq. (20):

$$\mathscr{F}_p = C - \frac{(\hat{v}-\mu)^2}{2\sigma^2}, \tag{21}$$

where

$$C \equiv h_0 + \sum p_j\left[\sqrt{\frac{|\Delta\mathbf{I}(j\Delta v)|}{\lambda}} - \frac{(j\Delta v - \hat{v})^2}{2\sigma^2}\right]. \tag{22}$$

Note that C is constant in the M step as the distribution $\{p_j\}$ is determined in the E step.

The center of mass has the advantage that it integrates over all the image evidence on the measurement line (unlike the strongest feature on the measurement line).

For the case of multiple sample points, the EM iteration is as follows:

E For all sample points on the contour, estimate the centers of mass.
M Minimize the sum of squared distances between sample points and corresponding centers of mass.

This algorithm implicitly assumes that the deformations ε at distinct sample points are statistically independent. This assumption depends of course on the distance between sample points. However, in general, if prior knowledge is available about statistical dependencies between deformations, then this knowledge should be incorporated into the shape model, i.e., the model should be a deformable template with parameterized deformations. In this case, the independent deformations at each sample point must be considered as additional to the parameterized deformations.

In the EM Contour algorithm, the centers of masses are used for minimizing the distance to sample points and therefore the algorithm resembles feature-based methods. The main difference lies in the distance measure to be minimized. In common feature-based methods the distances on the measurements lines are measured between contour and "'strongest features," while in the EM contour algorithm the distances are measured between contour and centers of probability mass. This is similar to the difference between MAP estimates and Bayes Least Squares estimates of the location of the object boundary. Performance comparisons between feature-based and the EM Contour method is given in [32].

6. Tracking results

The contour model is initialized on a fixed position and size using 100 samples. Σ_0 is set manually as to obtain a sufficient accuracy while still being able to allow some freedom of head movements. In the tests, the diagonal elements of Σ_0 are set to $Diag(\Sigma_0) = [\omega_c^2, \omega_c^2, \omega_\lambda^2, \omega_\lambda^2, \omega_\theta^2]^{\mathrm{T}}$. The values used in the experiments are given in Table 1.

The extent of the noise model, Σ, is initially set to twice Σ_0 and is then decreased back to Σ_0 after 10 frames. Each particle (ellipse) uses 20 equidistantly sampled normals to obtain the observations.

The mean of the sample set is optimized until convergence over 3 image scales with a maximum of 4 iterations of the EM contour algorithm on each image scale, resulting in a total of 12 iterations of the EM contour algorithm per frame. The use of the scale space approach significantly improves performance, especially in large images. As evaluating a single iteration in the EM contour method correspond to an evaluation of a particle, the additional cost required by applying the EM contour method in the image scale space is equivalent to adding 12 particles in the particle filter.

Initialization of the model is performed by asking the user to orient the head as to place the image of the eye within a rectangle displayed on the screen.

The method is tested using a 1.6 GHz PC with 128 Mb RAM on both Europeans and Asians in live test situations and prerecorded sequences using web and video cameras. In images of sizes 720 × 576 (digital video cameras) a frame rate of 25 frames per second is obtained.

In Figs. 7–11, images from testing the method on iris tracking using a standard video and web camera in both indoor and outdoor situations are shown. These images indicate that the method is capable of tracking the iris under scale changes, squinting the eye in various light conditions, under image defocusing, and under moderate head movements. Fig. 9 shows a sequence in which a video camera is used with built-in IR light emitter that allows to switch between visible-light and IR images. Changing between the two modes results in a significant change in image gray scale values. Despite these drastic changes, tracking is maintained without changing the model or any of its parameters.

Table 1
Parameters of the tracker

ω	16	Dimensionless[a]
ω_λ	0.32	Dimensionless[b]
ω_θ	1.1	Degrees

[a] Translation relative to the initial size of the iris.

[b] Scaling factor relative to the initial size of the iris.

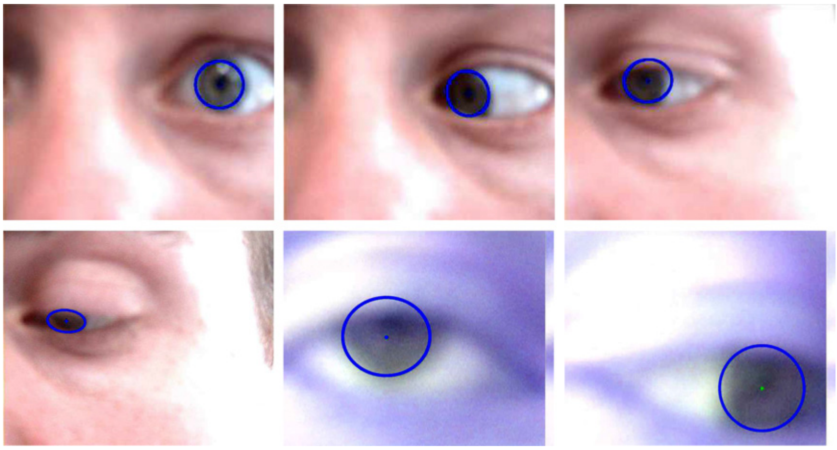

Fig. 7. Tracking the iris under various light conditions, head poses, image blurring, and scales.

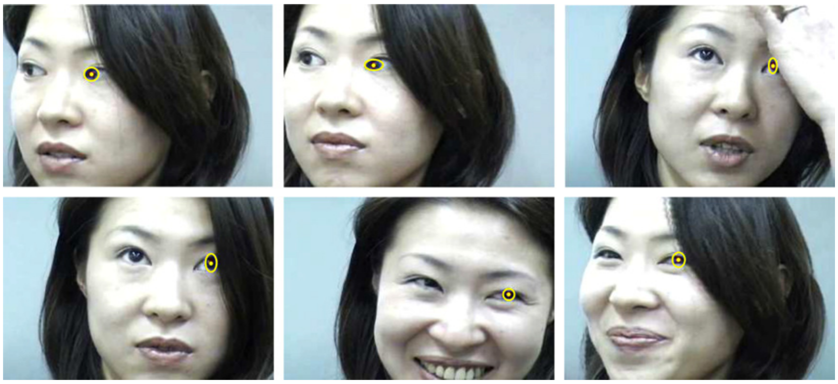

Fig. 8. Tracking of a Japanese eye with distractions from hair and hand.

The differences in Asian and Caucasian eyes seem not to significantly influence the tracking performance. Due to the changes in image quality, there is a vast difference in difficulty of tracking eyes in images using web cameras to using video cameras and IR-based images. However, the method is capable of tracking the iris without changing the model for all three types of images. Clearly, tracking accuracy improves with the changed image quality.

The number of particles is an important parameter in particle filtering. Too few particles result in insufficient accuracy while using too many particles wastes computation time. Fig. 12 shows the accuracy (measured in Euclidean distance, in pixels) for estimates of (a) the center and (b) the radius of the iris, as the number of particles change. The measurement of accuracy is based on a set of 2700 manually annotated images. Using a low number of samples results in low accuracy, but accuracy quickly

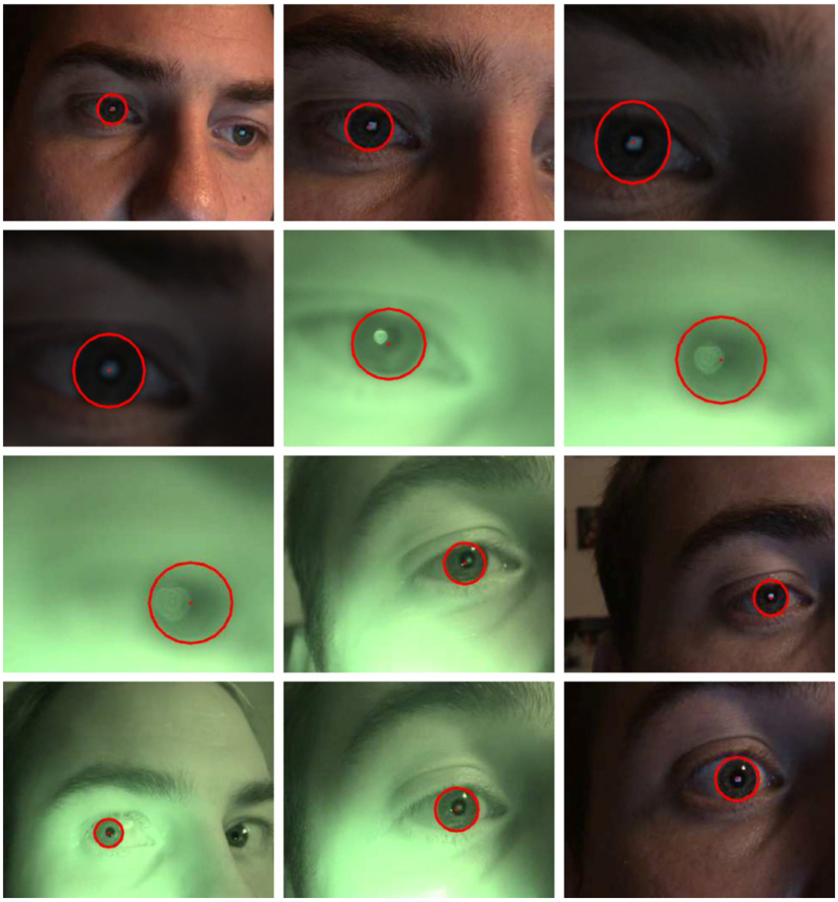

Fig. 9. Tracking under scale changes while illumination is changed between IR to non-IR lighting and under image defocusing. Notice how light conditions change when switching between IR and non-IR light emission.

improves with increasing sample size. Convergence is reached around 80–100 particles.

Fig. 12 also shows the mean accuracy (defined as above) for estimates of the iris center and radius as a function of the iris size in the image plane. Accuracy is quite high for low scales and decreases with the size of the iris; however the *relative* accuracy (relative to the iris size) actually increases with iris size and therefore better gaze estimates are possible when the iris size is larger.

6.1. Blink detection

Detection of eye blinks is a piece of information that can be used for various purposes, for example in user interaction. A simple approach to blink detection is

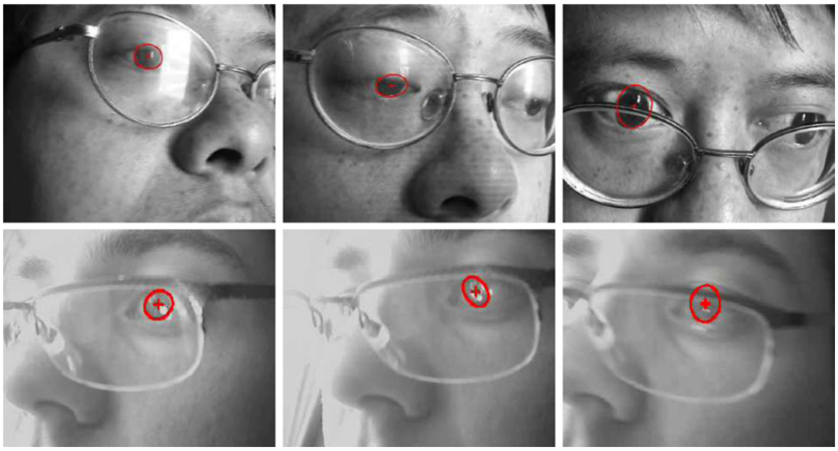

Fig. 10. Iris estimates of two persons wearing glasses on which light is reflected on the glass.

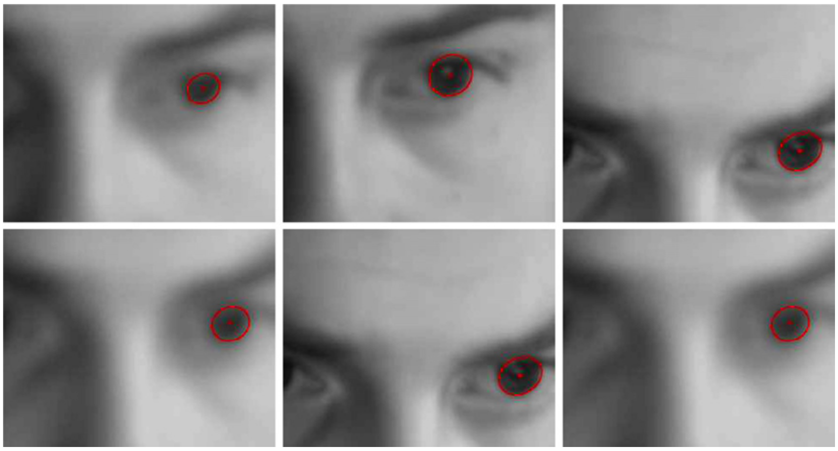

Fig. 11. Iris estimates under heavy blurring of the images due to image defocusing.

to detect that the iris disappears. In the case of contour models, the likelihood ratio in Eq. (14) expresses exactly what is needed, namely the ratio between the likelihood for the hypothesis that the target is present (from Eq. (13)) and the null hypothesis that the contour is not present (Eq. (7)).

In the following we use the likelihood ratio at convergence of the EM contour algorithm for detecting eye blinks. Obviously it is difficult to distinguish between tracking failures and blinks since the iris is not visible by the tracker in either case. However, Table 2 shows that 92.8 percent of 383 eye blinks were detected correctly under minor head movements.

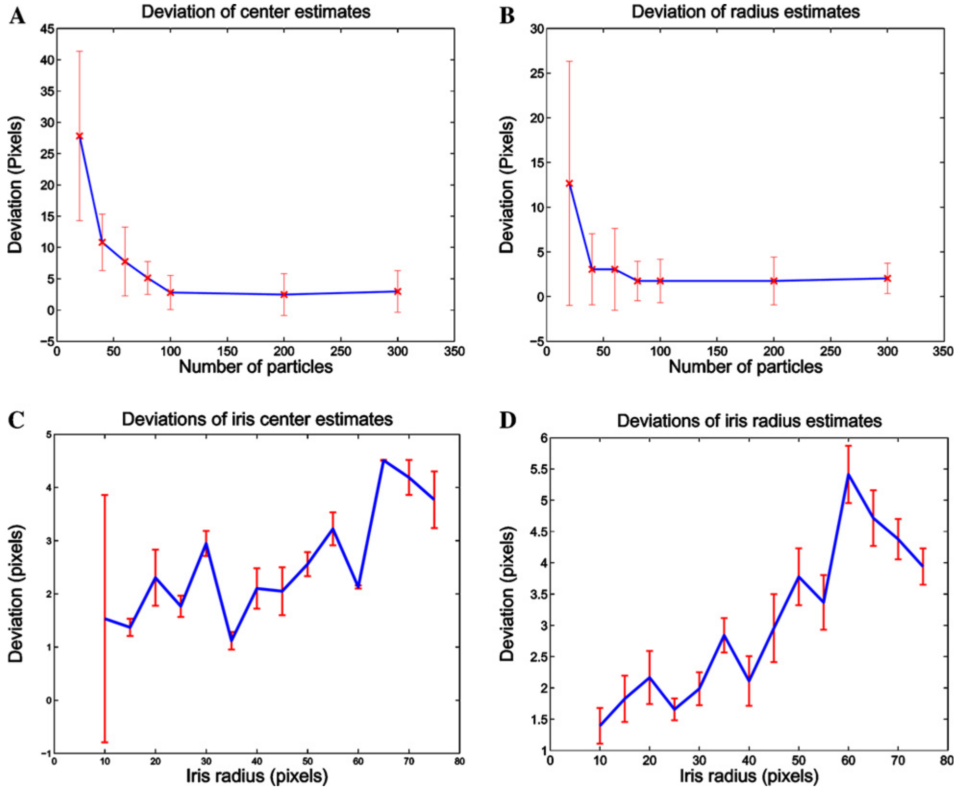

Fig. 12. Mean distance and error bars of (A) the centers and (B) radii as a function of the number of samples and (C) centers and (D) radii as a function of iris radius.

Table 2
Results of blink detection: 92% of 383 blinks are correctly detected

	Occurrences	Percent of total (383 blinks)
False positives	18	4.7
False negatives	11	2.8
Correctly detected	372	92.8

7. Gaze estimation

For the purpose of gaze estimation, we need to infer the point where the subject is looking given the image data. More specifically, we aim at finding the distribution $p(\mathbf{g}_S|\mathscr{I})$, where $\mathbf{g}_S$ is the gaze position (i.e. screen coordinates) and $\mathscr{I}$ is the image. The maximum a posteriori, maximum likelihood, or least-squares estimates are most often used and hence a deterministic mapping, $\Phi : \mathbb{R}^n \rightarrow \mathbb{R}^3$ from an image with n pixels to world coordinates is inferred. When using the gaze information for screen-based applications, the image of Φ is a subset of $\mathbb{R}^2$, as the depth is implicitly given. Thus only the mapping $\Phi : \mathbb{R}^n \rightarrow \mathbb{R}^2$ will be considered here. The process of gathering data for finding the transformation Φ is called *calibration*. Calibration is usually performed by the user looking at N calibration points $\mathbf{t}_i$ on the screen, while relating these to the image of the eye $\mathbf{x}_i$. A pair of camera-image coordinates $\mathbf{x}_i$ and calibration point coordinates $\mathbf{t}_i$ are called *conjugate*.

Several approaches have been suggested to determine gaze. As for eye-tracking methods, gaze-estimation methods can be divided into feature-based and appearance-based. Feature-based methods use features such as contours and eye corners for gaze determination. Due to the low number of features used, the size of the input space is reduced from the total number of pixels. IR-based eye trackers generally use feature-based methods as the center of the eye and the glint (reflection) are easily obtained [15,12,5,27]. Usually assuming a static head, methods based on this idea use the glint as a reference point: the vector from the glint to the center of the pupil describes the gaze direction. Wang and Sung [44] model the iris contours as two planar circles and estimate projections onto a retinal plane. Using the ellipsoidal shape of projected eyes, anthropometric knowledge and the known distance to the subject for gaze determination they achieve a 0.5 degree error. Matsumoto et al. [24] presents an eye gaze estimation method in which the eye corners are located using a stereo vision setup and a $3D$ head model. The eyeball position is calculated from the pose of the head and a $3D$ offset vector from the mid-point of the corners of an eye to the center of the eye. Beymer and Flickner [1] and Shih and Liu [35] studies the possibility of eliminating the need for a gaze calibration procedure by using two cameras and by exploiting the geometry of eyes and their images. Ji and Zhu [16] present a technique based on IR for gaze tracking in which no calibration is necessary; it allows for natural head movement and is completely non-intrusive while still producing relatively robust and accurate gaze tracking (about 5°) by using a neural network-based learning.

The appearance-based methods do not explicitly extract features, but use all the image information as input. Therefore, the dimensionality of the input space is much higher than feature-based methods. Pomerleau and Baluja [33] use 2000 cropped images of the eyes as input in a multi-layer neural network. Allowing for some head movements, an accuracy of 1.5° is obtained. A similar approach is taken by Xu et al. [47], who achieve a comparable accuracy, but with 3000 training samples. Tan et al. [38] use a nearest neighbor and linear interpolation in an IR appearance-based method. They use locally linear embedding [34] of the view manifold. 252 samples images of 20×20 pixels are needed to obtain an accuracy of 0.38 degrees using a leaving-one-out test. Using facial models for eye tracking may improve tracking performance and may be used for error correction, but may result in less accuracy in gaze estimation as the resolution of the eye is lower.

There does not seem to be a correlation between the number of calibration points used for different methods and the obtained accuracy. This variation may be due to the amount of prior information on the geometry and camera parameters, or by the different modalities i.e. visible vs. IR light. Using IR light emitters and a calibrated camera, but otherwise not explicitly specifying the geometry of the monitor, camera and user, both Ohno and Mukawa [30] and Villanueva et al. [42] have shown that 2 calibration points are sufficient. In most cases, the number of calibration points needed is found empirically. Using visible light and a calibrated camera, Wang et al. [45] proposed a method for obtaining gaze direction with 4 calibration points, but do not give a direct reason for their choice. The following section may justify their choice. It is possible by having sufficiently many calibrations points to sample the output space densely and thus good performance ought to be obtained, but requiring many points for calibration is tiring and inconvenient for the user. In the following section we present a lower bound on the number of calibration points when only the position of the center of the iris is known.

7.1. A lower bound on calibration points

In this section, we obtain a lower bound on the number of calibration points needed for gaze-based interaction using uncalibrated cameras and unknown, but fixed, setup geometry.

This lower bound is based on an approximation valid for the range of gaze directions of practical interest.

Modelling the eye as a sphere and assuming fixed head position, the position of the iris is defined by two rotation angles α, β of the eye for the horizontal and vertical directions. We further define the origin $\alpha = 0$, $\beta = 0$ as the position of the eye fixating the center of the screen.

Consider the distances a between a corner of the screen and the center of the screen, and b between the eye and the screen. The maximum value for the angle θ between the origin and the current direction of gaze is $\theta_M = \arctan(a/b)$ Typical values are $a \approx 23$ cm, and $b \approx 60$ cm, and therefore $\theta_M \approx 0.363$ radians. This is assuming that, when fixating the center of the screen, the optical axis of the eye is perpendicular to the screen: if the screen is tilted, then θ_M becomes even smaller.

Consider the plane E tangent to the eyeball at the point $\alpha = 0$, $\beta = 0$ (Fig. 13). Again, we define a coordinate system in this plane with the origin at the point tangent to the eyeball. Each direction of gaze (α, β) corresponds to one and only one point $\mathbf{e}$ on the E plane. It is clear from Fig. 13 that the point $\mathbf{e}$ and the point on the screen that is being fixated are related through a homography which we define as T_E^S. Defining r as the eye radius, it is also clear that $|\mathbf{e}| = r \tan\theta$.

The distance between iris and E plane is equal to r $(1 - \cos\,\theta)$ and therefore it is never larger than $r(1 - \cos\,\theta_M) \approx 0.065\ r$. This distance can be neglected when the camera image plane is almost parallel to the E plane. Therefore, we can consider that the camera is imaging the orthographic projection $\mathbf{e}'$ of the iris onto the E plane (see Fig. 13). Clearly, $|\mathbf{e}'| = r \sin\theta$ and therefore the relative error $2|\mathbf{e} - \mathbf{e}'|/|\mathbf{e} + \mathbf{e}'|$ is at most equal to the relative difference

$$2(\tan\theta_M - \sin\theta_M)/(\tan\theta_M + \sin\theta_M). \tag{23}$$

Inserting $\theta_M = 0.363$ into the above expression, the relative error can be seen to be at most equal to 0.068. Multiplied by the distance a on the screen, this corre-

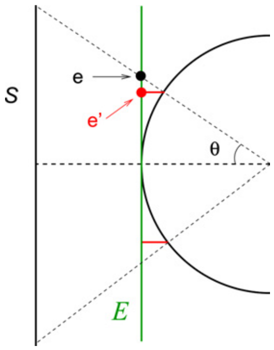

Fig. 13. The geometry used to derive a lower bound on the number of calibration points. The eye is represented by the hemicircle on the right-hand side, looking at the screen *S*. The tangent plane *E* is represented by a green line. Perspective projections of the center of the iris onto the *E* plane and onto the screen are represented by black dashed lines; orthographic projections of the iris center onto the *E* plane are represented by red line segments.

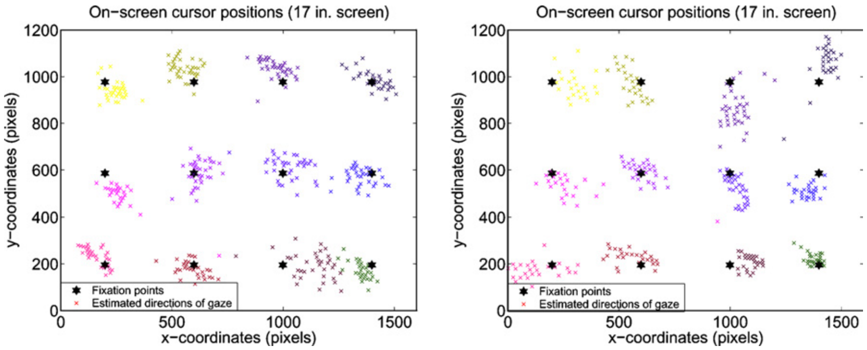

Fig. 14. Gaze Estimation on a 17 in. screen with a resolution of 1600 × 1200 pixels. The black stars represents a fixation point and the colored crosses shows the estimated gaze (each color is linked to a different fixation point).

sponds to an error of at most 1.56 cm. This systematic error is comparable to the random error in gaze estimation (see Fig. 14). Neglecting this error, we can assume that the camera is imaging $\mathbf{e}$, instead of $\mathbf{e}'$. Therefore, there is an approximate homography from the E plane to the camera image plane. We define this homography as T_C^E.

The concatenation of two homographies is a homography and therefore the transformation from image to screen coordinates via the eye, $\Phi = T_C^S = T_C^E \circ T_E^S$, is a homography. A homography is defined by 4 points and hence the transformation from image to screen coordinates is defined by at least 4 points. If the head moves together with the eyes when the gaze is shifted on the screen, additional calibration points are needed; thus, 4 points can only be considered a lower bound.

To summarize, we have proven that four calibration points are sufficient if the following approximations are valid:

1. the eye is spherical;
2. the head is fixed;
3. the maximum distance between iris and E plane is negligible from the viewpoint of the camera;
4. the maximum distance between points $\mathbf{e}$ and $\mathbf{e}'$ is negligible.

7.2. Gaze estimation results

In these experiments the setup between the user, camera (including the internal camera parameters) and monitor is unknown. The users are sitting about 40–60 cm away from the monitor. Using four calibration points an accuracy of 4 degrees is obtained. Fig. 14 shows the results of two sessions in which the users fixate the gaze on 12 predefined points on the screen (displayed from top left to bottom right). The errors are calculated under the assumption that the user looks at the center of the fixation point on the screen. On average the systematic errors are 1.27 and 0.76 cm in x and y directions respectively. The superimposed random errors are 3.56 and 1.27 cm in x and y directions respectively.

Fig. 15 shows the accuracy over the sessions in Fig. 14. The two large peaks in the plots correspond to the user still looks at fixation point on the left-hand side of the monitor while the actual fixation point has m Both exhibit an increase in deviation after about 200 frames. These deviations are likely to be due to changes of head positions.

It is instructive to consider how a small deviation in iris location propagates to errors in gaze estimation. Let δ_I denote the error associated with estimating the center $\mathbf{x}_I$ of the iris, Φ the estimated homography from the image to the screen and $\mathbf{t}_M$ the point the user looks at. Ideally $\mathbf{t}_M = \Phi x_I$, but, due to the error in estimating the iris location, $\mathbf{t}_M + \delta_M = \Phi(\mathbf{x}_I + \delta_I)$, where $\delta_M = \Phi\delta_I$. Letting $\delta_I = 4$ pixels in the image (as obtained from Fig. 12) the mean value of δ_M over the test set is calculated to be about 64 pixels (corresponding to 1.22 cm on the monitor). It seems that the er-

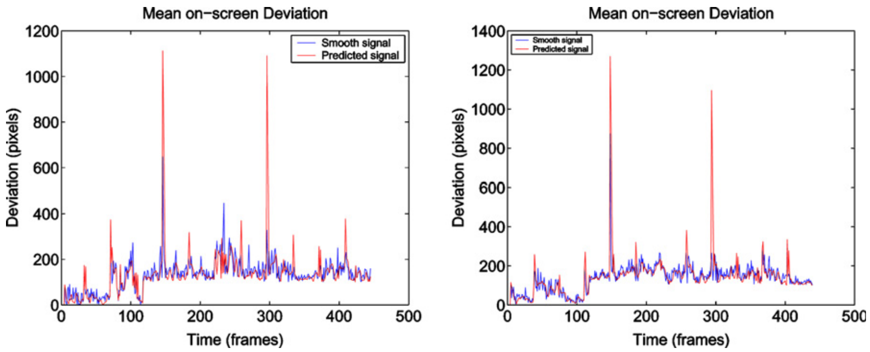

Fig. 15. Errors in gaze estimation over time (frames). The two peaks of the errors correspond to the two cases in which the user still gazes on, i.e., the right while the actual point being displayed is on the opposite side. A version of the signal smoothed over the previous two frames is also shown.

rors induced by changes of head position are likely to be larger than those implied by the iris tracker (as shown in Fig. 15).

8. Discussion

A tracking method based on particle filtering and the EM Contour algorithm is described and applied for iris tracking. The method has proven robust for tracking eyes under moderate head movement, image defocusing, occlusions, and illumination changes, thanks to the avoidance of "feature detection." Thus, the method is well suited for both high quality and low cost eye tracking. To perform gaze estimation with this method, a lower bound on the number of calibration points needed for gaze estimation is presented and its application is demonstrated. The lower bound applies in situations where the setup is unknown, but fixed: it does not assume any camera calibration or the user sitting in a particular distance from the screen. However, it does assume that the user does not move the head. This is a somewhat restricted scenario as head movements cannot be prevented in all situations. Allowing for head movements increases the dimensionality and thus the lower bound still applies.

The ellipse model is a good approximation of the iris shape. Building shape models of the iris or the entire eye is a simple extension of the method explained. A more committed model (i.e., a full shape model of the eye) may provide additional robustness to the eye tracker. However, the main purpose of this paper is to show that explicit feature detection is not needed for the purpose of iris tracking. How the method works for more elaborate models and tracking tasks remains to be investigated.

References

[1] D. Beymer, M. Flickner, Eye gaze tracking using an active stereo head, in: IEEE Computer Vision and Pattern Recognition (CVPR), vol. II, 2003, pp. 451–458.

[2] T.F. Cootes, C.J. Taylor, Active shape models—'smart snakes', in: Proc. British Machine Vision Conf., BMVC92, 1992, pp. 266–275.

[3] J. Coughlan, A. Yuille, C. English, D. Snow, Efficient deformable template detection and localization without user initialization, Comput. Vis. Image Understand. 78 (3) (2000) 303–319.

[4] J. Deng, F. Lai, Region-based template deformation and masking for eye-feature extraction and description, Pattern Recognit. 30 (1997) 403–419.

[5] Y. Ebisawa, S. Satoh, Effectiveness of pupil area detection technique using two light sources and image difference method, in: 5th Annual Int. Conf. of the IEEE Eng. in Medicine and Biology Society, 1993, pp. 1268–1269.

[6] G. Edwards, T.F. Cootes, C.J. Taylor, Face recognition using active appearance models, in: ECCV'98. 5th European Conf. on Computer Vision. Proc, vol. 2, Springer-Verlag, 1998, pp. 581–95.

[7] K. Grauman, M. Betke, J. Gips, G. Bradski, Communication via eye blinks: detection and duration analysis in real time, in: IEEE Computer Vision and Pattern Recognition (CVPR), vol. I, 2001, pp. 1010–1017.

[8] P. Hallinan, Recognizing human eyes, SPIE 1570 (1991) 214–226.

[9] D.W. Hansen, J.P. Hansen, M. Nielsen, A.S. Johansen, M.B. Stegmann, Eye typing using markov and active appearance models, in: IEEE Workshop on Applications on Computer Vision, 2003, pp. 132–136.

[10] R. Herpers, M. Michaelis, K. Lichtenauer, G. Sommer, Edge and keypoint detection in facial regions, in: Int. Conf. on Automatic Face and Gesture Recognition, 1996.

[11] J. Huang, D. Mumford, Statistics of natural images and models, in: IEEE Computer Vision and Pattern Recognition (CVPR), vol. I, 1999, pp. 541–547.

[12] T. Hutchinson Jr., K. Reichert, L. Frey, Human–computer interaction using eye-gaze input, IEEE Trans. Syst. Man Cybernet. 19 (1989) 1527–1533.

[13] M. Isard, A. Blake, Contour tracking by stochastic propagation of conditional density, in: Eur. Conf. on Computer Vision, 1996, pp. 343–356.

[14] J. Ivins, J. Porrill, A deformable model of the human iris for measuring small 3-dimensional eye movements, Mach. Vis. Appl. 11 (1) (1998) 42–51.

[15] Q. Ji, X. Yang, Real time visual cues extraction for monitoring driver vigilance, Lecture Notes Comput Sci 2095 (2001) 107.

[16] Q. Ji, Z. Zhu, Eye and gaze tracking for interactive graphic display, in: Proceedings of the 2nd International Symposium on Smart graphics, 2002, pp. 79–85.

[17] M. Kass, A. Witkin, D. Terzopoulos, Snakes: active contour models, in: Int. Conf. on Computer Vision, 1987, pp. 259–268.

[18] S. Kawato, N. Tetsutani, Detection and tracking of eyes for gaze-camera control, in: VI02, 2002, p. 348.

[19] K. Lam, H. Yan, Locating and extracting the eye in human face images, Pattern Recognit. 29 (1996) 771–779.

[20] LC Technologies, 2004. Available from: <http://www.eyegaze.com>. LC Technologies INC.

[21] G. Loy, A. Zelinsky, Fast radial symmetry for detecting points of interest, PAMI (2003) 959–973.

[22] J. MacCormick, A. Blake, A probabilistic contour discriminant for object localisation, in: Int. Conf. on Computer Vision, 1998, pp. 390–395.

[23] P. Majaranta, K.-J. Räihä, Twenty years of eye typing: systems and design issues, in: Symposium on ETRA 2002: Eye Tracking Research Applications Symposium, New Orleans, Louisiana, 2002, pp. 944–950.

[24] Y. Matsumoto, T. Ogasawara, A. Zelinsky, Behaviour recognition based on head pose and gaze direction measurement, in: IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS), 2000, pp. 2127–2132.

[25] Y. Matsumoto, A. Zelinsky, An algorithm for real-time stereo vision implementation of head pose and gaze direction measurement, in: Int. Conf. on Automatic Face and Gesture Recognition, 2000, pp. 499–504.

[26] B. Moghaddam, A. Pentland, Probabilistic visual learning for object representation, PAMI 19 (7) (1997) 696–710.

[27] C. Morimoto, D. Koons, A. Amir, M. Flickner, Pupil detection and tracking using multiple light sources, IVC 18 (4) (2000) 331–335.

[28] R. Newman, Y. Matsumoto, S. Rougeaux, A. Zelinsky, Real-time stereo tracking for head pose and gaze estimation, in: Int. Conf. Automatic Face and Gesture Recognition, 2000, pp. 122–128.

[29] M. Nixon, Eye spacing measurements for facial recognition, Appl. Digital Image Process. 575 (VIII) (1985) 279–285.

[30] T. Ohno, N. Mukawa, A free-head, simple calibration, gaze tracking system that enables gaze-based interaction, in: Eye Tracking Research and Applications Symposium, 2004, pp. 115–122.

[31] A. Pece, A. Worrall, Tracking with the EM contour algorithm, in: Eur. Conf. on Computer Vision, vol. I, 2002, pp. 3–17.

[32] A.E.C. Pece, The Kalman-EM contour tracker, in: Proc. third Workshop on Statistical and Computational Theories of Vision: SCTV 2003, 2003. Available from: URL <http://department.stat.ucla.edu/~yuille/meetings/2003_workshop.php>.

[33] D. Pomerleau, S. Baluja, Non-intrusive gaze tracking using artificial neural networks, in: CMU-CS-TR, 1994.

[34] S. Roweis, L. Saul, Nonlinear dimensionality reduction by locally linear embedding, Science 290 (5500) (2000) 2323–2326.

[35] S.-W. Shih, J. Liu, A novel approach to 3-d gaze tracking using stereo cameras, IEEE Trans. Syst. Man Cybernet. 34 (1) (2004) 234–245.

[36] H. Sidenbladh, M.J. Black, Learning the statistics of people in images and video, Int. J. Comput. Vis. 54 (1–3) (2003) 183–209.

[37] Smart Eyes, 2004. Available from: <http://www.smarteye.se>. Smart Eyes A/B.

[38] K.Tan, D. Kriegman, N. Ahuja, Appearance-based eye gaze estimation, in: Workshop on Applications of Computer Vision, 2002, pp. 191–195.

[39] P. Tissainayagam, D. Suter, Tracking multiple object contour with automatic motion model switching, in: Int. Conf. on Pattern Recognition, 2000.

[40] Tobii, 2004. Available from: <http://www.tobii.se/>. Tobii Technologies.

[41] A. Tomono, M. Iida, Y. Kobayashi, A tv camera system which extracts feature points for non-contact eye movement detection, in: SPIE Optics, Illumination, and Image Sensing for Machine Vision., vol 1194, 1989, pp. 2–12.

[42] A. Villanueva, R. Cabeza, S. Porta, Eye tracking system model with easy calibration, in: Eye Tracking Research and Applications Symposium, 2004, pp. 55–55.

[43] P. Viola, M. Jones, Robust real-time face detection, in: Int. Conf. on Computer Vision, vol. II, 2001, p. 747.

[44] J. Wang, E. Sung, Gaze determination via images of irises, Image Vis. Comput. 19 (12) (2001) 891–911.

[45] J.-G. Wang, E. Sung, R. Venkateswarlu, Eye gaze estimation from a single image of one eye, in: Ninth IEEE Int. Conf. on Computer Vision, 2003, pp. 136–143.

[46] D.J. Ward, A.F. Blackwell, D.J.C. MacKay, Dasher: a gesture driven data entry interface for mobil computing, Hum Comput. Interact. 17 (2002) 199–228.

[47] L. Xu, D. Machin, P. Sheppard, A novel approach to real-time non-intrusive gaze finding, in: British Machine Vision Conference, 1998.

[48] J. Yang, R. Stiefelhagen, U. Meier, A. Waibel, Robust detection of facial features by generalized symmetry, in: Int. Conf. on Pattern Recognition. vol. I, 1992, pp. 117–120.

[49] D. Young, H. Tunley, R. Samuels, Specialised hough transform and active contour methods for real-time eye tracking. Tech. Rep. 386, School of Cognitive and Computing Sciences, University of Sussex, 1995.

[50] A. Yuille, J. Coughlan, Fundamental limits of Bayesian inference: Order parameters and phase transitions for road tracking, IEEE Trans. Pattern Anal. Mach. Intell. 22 (2) (2000) 160–173.

[51] A.L. Yuille, P.W. Hallinan, D. Cohen, Feature extraction from faces using deformable templates, Int. J. Comput. Vis. 8 (2) (1992) 99–111.

[52] Z. Zhu, K. Fujimura, Q. Ji, Real-time eye detection and tracking under various light conditions, in: Symposium on ETRA 2002: Eye Tracking Research Applications Symposium, New Orleans, Louisiana, 2002a, pp. 139–144.

[53] Z. Zhu, Q. Ji, K. Fujimura, Combining kalman filtering and mean shift for real time eye tracking, in: Int. Conf. on Pattern Recognition, vol. IV, 2002b, pp. 318–321.