

Statistical approaches to eye extraction and tracking

T.S Nedzamba, F-P Clouet, B.J. van Wyk, M.A. van Wyk

French South African Technical Institute in Electronics
Private Bag X680 Pretoria, 0001. Republic of South Africa. Tshwane University of Technology,
Tel: +27 12 382 4191 Fax: +27 12 382 5294

thendo.nedzamba@fsatie.ac.za, fpclouet@yahoo.fr, vanwykb@tut.ac.za

Abstract — *In this paper a novel statistical method is proposed for the detection of iris centres for the purpose of eye tracking. The detection of an iris is preceded by a statistical method to detect the face and a gradient map projection is used to locate eye regions. Reduced computational cost and improved performance is a common objective of researchers in the field of eye tracking. The purpose of this paper is to demonstrate the ability of statistical techniques to detect a face and track the eyes in complex backgrounds with different skin colour types.*

Keywords: *Face detection, Eye extraction, Statistical methods, and Eye tracking.*

1 INTRODUCTION

Eyes are distinct facial features. Many algorithms have been proposed in an attempt to achieve high accuracy and robustness in the detection of eyes. Many of these methods depend on measuring infrared light reflected on the cornea. Infrared cameras are used when acquiring video frames of the subject. There are also other methods that use standard cameras. Three of the more well-known eye detection methods are reviewed in the following paragraphs.

It has been noted in the experiments conducted by Perez *et al.* [4] that at least four illuminators of a specific shape should be used to accurately detect and validate a single infrared glint. This method is not cost effective and not easy to implement. In an attempt to achieve robustness in the presence of head movement, Perez *et al.* [4] used two cameras for a narrow field of view and two cameras for a wide field of view.

Douglas *et al.* [1] proposed a method to extract eye contours based on computing peak and valley maps of intensity images. It has been shown that eye corners could accurately be detected using this method, provided that the head is at a fixed position.

Peng *et al.* [2] proposed a combination of feature-based and template matching methods to accurately estimate the position of the iris. In their approach they used feature-based techniques to detect eye regions, and template matching was used to search for iris centres in a reduced area of the image. However, in some cases the template matching process has to be repeated several times to accommodate different eye sizes in an input image.

In order to improve the proposed template matching, peak-valley and feature-based methods, a more robust algorithm working with a *standard camera* and complex backgrounds, is proposed. In our experiments, a combination of eye centres and pupil radii were used as measurement vectors for a tracking filter. The following processing steps are implemented, Face detection (section 2.1), Eye region extraction (section 2.2), Eye detection (section 2.3), and Eye tracking (section 2.4)

Different iris radii are allowed in order to accommodate different eye dimensions. To obtain training data, pupil radius measurements of 20 people were conducted using a fixed camera at a fixed distance. A Bayesian method, originally proposed for lip tracking [5], but modified for our application, has been implemented. This method can be used to model the shape of an image using boundary and texture content [5]. The most crucial part of this

Bayesian method is to precisely define a prior distribution.

2 PROCESSING STEPS

2.1 Face Detection

This section is based on a study done by Chang and Robles [8]. There are various ways to detect the face of a subject. Eyes can be more easily tracked when a subject's face has been properly localised. In this section only a skin colour method for localising the face is considered. This method is used for localising the highest region of interest, assumed to be the area containing the subject's face. The algorithm detects the face irrespective of non-skin background colour changes, making it suitable for varying background applications.

2.1.1 Statistical colour model

This method is based on detecting skin colour pixels in the captured image. It is likely to detect an area in the image that has skin colour pixels, but is not part of the subject's face. Blob size algorithms are used to reject these regions. The first thing to do is to extract the skin region from the image. Skin colour pixels have similar colour but may differ in intensity. Since brightness is not important when representing human skin colour, a normalized chromatic colour space representation can be used [8] given by

$$r = R/(R+G+B) \quad (1)$$

$$b = B/(R+G+B) \quad (2)$$

The corresponding r and b values are obtained for each pixel (i, j) , if these values are greater than a certain threshold (chosen empirically) the pixel (i, j) represents the skin pixels, otherwise it represents non skin pixels.

The skin colour pixels are found in the same chromatic space region and are modelled using a Gaussian distribution (Figure 2), $N(\mu, S)$ where μ is the mean vector of the two chromatic components (r, b) and S is their covariance. Here $\mu = (\mu_1, \mu_2)$

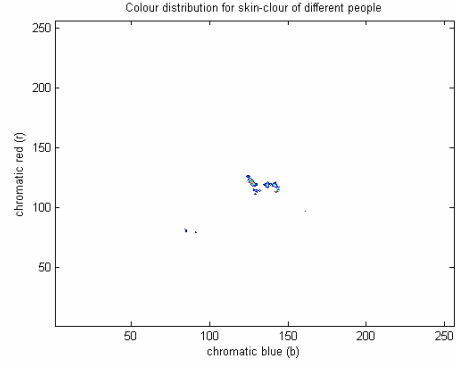


Figure 1: Skin-colour clustered in chromatic space (red and blue on the Y and X axis respectively, green is omitted).

where

$$\mathbf{m}_1 = \bar{r} = \frac{1}{N} \sum_{i=1}^N r_i, \quad (3)$$

and

$$\mathbf{m}_2 = \bar{b} = \frac{1}{N} \sum_{i=1}^N b_i \quad (4)$$

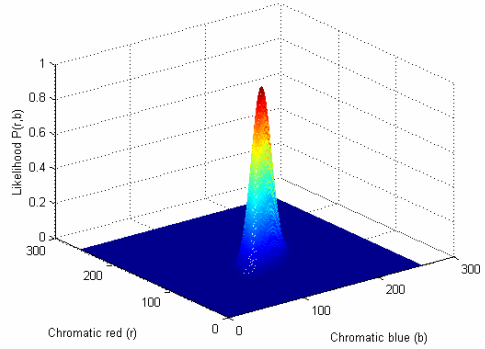


Figure 2: Skin-colour Gaussian model

Skin images were sampled to obtain a representative colour distribution of human skin in chromatic space. Twenty images of different persons were used combined with low-pass filtering to reduce noise. Figure 1 shows the colour distribution of these skin samples in the chromatic colour space. The likelihood of skin colours (r, b) in chromatic space is given by

$$P(x) = \exp\left(-\frac{1}{2}(x-\mathbf{m})^t \Sigma^{-1}(x-\mathbf{m})\right) \quad (5)$$

where $x = \{r, b\}$ and $\mathbf{m} = (\mathbf{m}_1, \mathbf{m}_2)^t$.

The image with detected skin-like pixels is then binarized and a 'blob' operation is performed to detect the face region. The value of each pixel

represents its likelihood to belong to the face and hence a threshold can be used for discrimination. Figure 3 shows an example of an input image and a detected face.

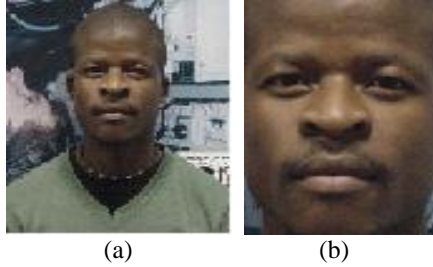


Figure 3: (a) Input image and (b) detected face

2.2 Eye region extraction

The eye region can be extracted once the face has been detected. This region is found by computing the gradient map of the detected face,

$$G(r, c) = |I(r, c) - I(r-1, c)|, \quad (6)$$

as shown in Figure 4, where r and c are row and column indices and I is the face image.

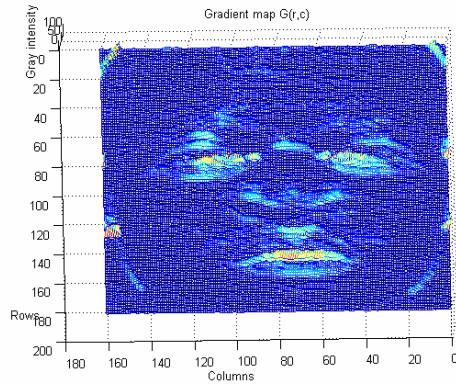


Figure 4: Gradient map

Gradient maps are useful for detecting regions of high spatial frequency using a projection approach. In our case to find the eye region, the projection of the gradient $G(r, c)$ on its vertical axis is computed; this is done by calculating $H(r)$ defined by

$$H(r) = \sum_{i=1}^X G(r, i), \quad (7)$$

where X is the length of the row vector. The result of the projection of the gradient in the

figure 4 is showed in Figure 5. The gradient map is projected horizontally to localise a region that contains eyes

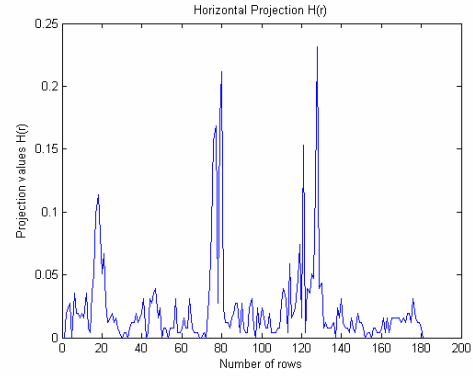


Figure 5: Horizontal projection

The maximum value of the horizontal projection (i.e. for $H(r) < \text{median value of rows}$, Figure 5) is then used to determine the eye region in the image (Figure 6). The region containing both eyes is reduced by dividing symmetrically to obtain a single eye region. Working with single eye regions is computationally more efficient and eliminates unnecessary features of the face.



Figure 6: Detected eye region

2.3 Eye Detection

This section describes our proposed method to extract the iris centres using model parameters and a statistical method. The quality of the image plays an important role in the accuracy of this method since it depends on the intensity matrix. The geometric template of the iris is given by the parameters of a circle. The iris is modelled using a simple two-dimensional template,

$$(x - x_c)^2 + (y - y_c)^2 = r^2, \quad (8)$$

where (x_c, y_c) are the centre coordinates of the iris and r its radius.

2.3.1 The Prior function

From the test data of 20 subjects the fitted radius distribution is $r = N(7, 0.4)$ and on average the

iris centre is at $(x_c, y_c) = (34, 25)$, using fixed-size eye region images of 40×80 pixels.

The joint prior of (x, y) and r are then obtained using the probability density function (pdf) of the normal distribution with mean and variance ($\mathbf{s}^2 = 0.4$) as given by

$$f(x, y, r) = \exp\left(-\frac{1}{2(0.4)}(r-7)^2\right) \quad (9)$$

For $0 < x < M$ and $0 < y < M$, otherwise $f(x, y, r) = 0$

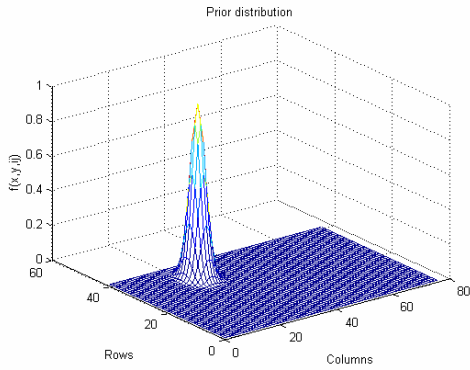


Figure 7: Prior function

2.3.2 Likelihood function

The likelihood function is devised to relate the measurement to an iris template (see Equation 8). This function depends on lighting variations. To model the underlying pixel intensities, a statistical model similar to that used by Shah and Marshall [6] is proposed, namely

$$L(I | \mathbf{q}) = \frac{1}{Z} \exp\left(-\frac{1}{2} \sum_k E_k\right). \quad (10)$$

Here E_k is the energy associated with the k^{th} circle template region in the sub-image and Z is a normalizing constant. The energy function is therefore

$$E_k = \frac{1}{\mathbf{s}_k} \sum_{j \in k} (I_j - \mathbf{m}_k)^2, \quad (11)$$

where $(\mathbf{m}_k, \mathbf{s}_k) = (7, 0.4)$ are the parameters modelling the energy of the k^{th} region.

2.3.3 The Posterior Function

To make inferences the posterior distribution $\mathbf{p}(\mathbf{q} | I)$ must be obtained using Bayes rule, i.e.

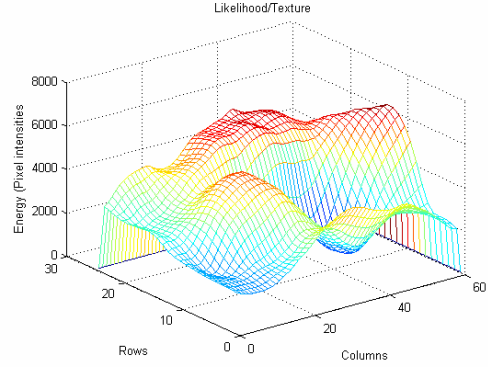


Figure 8: Likelihood/Texture function

$$\mathbf{p}(\mathbf{q} | I) \propto \exp\left(-\frac{1}{2} E_k\right) f(x, y, r). \quad (12)$$

From this posterior distribution, the valleys (minima) of the Energy/Intensity regions can be obtained. The deepest valley represents the iris centre. The Metropolis-Hastings algorithm, described in the next section, is used to find the valleys.

2.3.4 Sampling from the complex distribution

It is not trivial to sample from a distribution whose form is non-standard, but fortunately the Metropolis-Hastings algorithm provides a methodology to do this.

To generate random numbers from a non-standard distribution, Markov chain algorithms draw dependent samples whose stationary distribution is the posterior distribution. Random numbers are generated as follows:

1. Assume an initial value $x(t)$.
2. Draw a sample, $y(t)$ from a proposal distribution $q(y | x(t))$.
3. Accept $y(t)$ as the next sample $x(t+1)$ with probability $r(x(t), y(t))$, and keep $x(t)$ as the next sample $x(t+1)$ with probability $1 - r(x(t), y(t))$, where

$$r(x, y) = \min \left\{ \frac{f(y) q(x|y)}{f(x) q(y|x)}, 1 \right\} \quad (13)$$

4. Increment $t \rightarrow t+1$, and repeat steps from 2 to 4 until the desired number of samples are obtained.

The sample with the lowest value is chosen as the iris centre.

2.4 Eye tracking

In order to track eye features in subsequent frames, we used the Kalman filter. There are a couple of methods that can be used to tackle this issue, but Kalman filtering is computationally efficient, is a recursive procedure, and requires a minimum amount of storage of previous samples [6].

The Kalman filter model represents both the information about the system and the measurement noise. A previous sample is used to predict the next state, which is updated when a new measurement becomes available.

The Kalman filter requires prior knowledge of the system. Our formulation for eye tracking closely follows the approach of Xie *et al.* [6] which can be consulted for more detail:

Let the state vector be $S = \begin{bmatrix} x_c \\ y_c \end{bmatrix}$. The system equation can be modelled as

$$s_{k+1} = s_k + w_k, \quad (14)$$

where k is the frame index and w_k is assumed to be a Gaussian white noise with zero mean, i.e. $w_k \sim N(0, Q_k)$. The measurement equation is given by

$$Z_k = H_k s_k + v_k \quad (15)$$

where $Z_k = \begin{bmatrix} x_k \\ y_k \end{bmatrix}$ is the measurement vector and H_k is the measurement matrix. Measurement noise v_k is modelled as white Gaussian noise with zero mean i.e. $v_k \sim N(0, R_k)$. The a priori state estimate for the time update s_{k+1}^- , a priori state error

covariance matrix P_{k+1}^- , Kalman gain matrix K_{k+1} , a posterior state error covariance matrix P_{k+1} , and a posterior state estimate s_{k+1} are computed as in Xie *et al.* [6].

3 DISCUSSION AND RESULTS

The algorithms discussed were implemented in MATLAB 7.0 on a Pentium 4 Windows NT workstation and were tested with images taken with a 1.3 mega pixel camera of 20 persons with different skin colour, hair colour, and also different iris radii. The texture of the skin, lighting conditions, and quality of the images played a significant role in the performance of both skin colour and eye models. From the experiment it was found that the algorithm used for eye detection required a precise formulation of the likelihood, i.e. the energy function. Figure 9 shows the detected eye centres (x_c, y_c) and the iris in different frames.

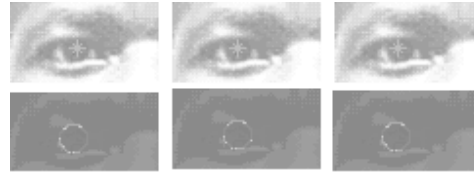


Figure 9: Detected eye centres and iris

Although the Kalman filter estimation algorithm is of a low order, the process model is sufficient for smooth tracking. Figure 10 shows the iris detected during the tracking process and Figure 11 shows the Kalman filter performance during the tracking process



Figure 10: Tracked eye positions

The Mean Squared Error (MSE) of 7.17 pixels obtained using the method proposed in this paper, makes the Kalman filter's accuracy and performance acceptable for the purpose of this work.

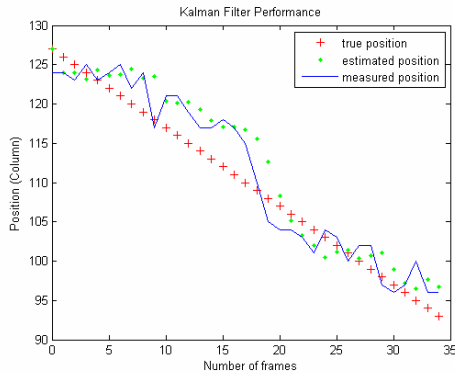


Figure 11: Kalman filter performance, true position is the measurement taken manually, estimated position is s_{k+1} , and measured position Z_k .

4 FACE AND EYE DETECTION COMPARISONS

Table 1

	TM	FB/TM	PV	Stats
<i>Accuracy</i>	65%	70%	55%	75%
<i>Robustness</i>	low	med.	low	high
<i>Speed</i>	fast	slow	med	slow

The comparison of the TM - Template matching, FB – feature-based, PV – peak-valley, and Stats – statistical methods proposed in this paper, are based on tests done using the same dataset. These comparisons only deal with the detection of a face and eyes. In Table 1 ‘low’ means not suitable for diverse backgrounds (restricted to iris radii and homogeneous skin colour), ‘med’ means that background restrictions apply, and ‘high’ implies usage with different backgrounds and skin types.

The statistical approach described in this paper is more robust to skin complexion variation and in general gives better detection results than the other three methods it was compared against.

5 CONCLUSION

In this paper a novel statistical method was proposed for the detection of iris centres for the purpose of eye tracking. This paper demonstrated

the ability of statistical techniques to detect a face and track eyes in complex backgrounds with different skin colour types. As with similar methods, lighting variations and poor quality images remain problematic.

Improvements can be made to detect the position of the eyes and the radius of the iris accurately in poor quality images: A template matching technique can be used to locate the best local minimum on the energy function.

6 REFERENCES

1. T.S. Douglas, F. Martinez, E.M. Meintjes, C.L. Vaughan, and D. Viljoen, “Eye feature extraction for diagnosing the facial phenotype associated with fetal alcohol syndrome”, *Medical & Biological Engineering & Computing*, vol. 41, pp. 101-106, 2003.
2. K. Peng, L. Cheng, S. Ruan, and G. Kukharev, “A robust algorithm for eye detection on gray intensity without spectacles”, *JCS&T* vol.5 no.3, pp 127-131, 2005.
3. T. D’Orazio, M. Leo, G. Cicirelli, and A. Distanto, “An Algorithm for real time eye detection in face images”, *Proceedings of the 17th International conference on pattern recognition*. Pp 1051-4651, 2004.
4. A. Pérez, M.L. Cordoba, A. Garcia, R. Mendez, M.L. Munoz, J.L. Pedraza, and F. Sanchez, “A precise eye-gaze detection”, *WSCG poster proceedings*, 2003.
5. D. Shah. And S. Marshall, “Image models for facial feature tracking”, In *nonlinear model-based image/video processing and analysis*, John Wiley and Sons, New York, pp. 299-314, 2001.
6. X. Xie, R. Sudhakar, H. Huang, “Real-time eye feature tracking from a video image sequence using kalman filter”, *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. 25, No. 12, pp. 1568-1577, 1995.
7. H. Veeraraghavan and N.P. Pananikolopoulos, “Detecting driver fatigue through the use of advanced face monitoring techniques”, *Center for transport studyReport* 01-05, 2001.
8. H. Cheng & U. Robles. <http://www-cs-students.stanford.edu>, accessed: 2005.

Acknowledgement: The DoD, DAS, TUT, and the NRF are thanked for their financial contributions towards this work.