2011 Eighth International Conference on Information Technology: New Generations

Eye-based HCI with full specification of mouse and keyboard using pupil knowledge in the gaze estimation

Kohei Arai and Ronny Mardiyanto
Saga University

Abstract

Eye-based Human Computer Interaction (HCI) with full specification of mouse and keyboard by human eyes only is proposed for, in particular, disabled person and for input device of wearable computing. It utilizes pupil knowledge in gaze estimation process. Existing conventional eye-mouse, gaze-mouse does not robust against various types of users with different features of color, shape and size of eyes and also is affected by illumination changes, users' movement (attitude changes), etc. Using knowledge about features, the influences are eliminated. Also the proposed eye-based HCI system allows simultaneous key input of more than three keys, for instance, "Ctl+Alt+Del" for initiating task manager, left/right click and drag/drop of mouse event capabilities. Although currently commercially available screen keyboard allows simultaneous key input for two keys such as "shit+", "Alt+*", it does not allow three keys input at once. Such these full specification of mouse events and keyboard functions are available for the proposed HCI system. Experimental results with six different nationalities of users with the different features shows effectiveness of using the knowledge for improvement of gaze estimation accuracy.*

Key Words: human computer interaction, gaze estimation, eye-mouse, keyboard operation, mouse event .

1. Introduction

The method for gaze estimation is divided into the following six categories by the method for eye detection and tracking with line of sight vector estimation [1]-[4];

(1) Limbus Tracking method [5]
(2) Purkinje method [6]
(3) EOG (Electro-occulography) method [7]
(4) Search coil method [8]
(5) Fundus Haploscope [9]
(6) Image analysis method [10]-[15]

The methods (1) to (5) do not impose users any psychological and physical burden because the methods require direct sensors attached to user face. Also the methods are relatively costly. On the other hand, the image analysis method does not insist any psychological and physical impact to users and is realized with comparatively cheap system. For this reason, we propose image analysis based method using gaze estimation (gaze is defined as the line which connects between eyeball center and pupil center). In gaze estimation, pupil location estimation is fundamental reference [17]-[27]. Pupil detection accuracy is very important because gaze estimation accuracy depends on this. Many researchers did not pay much attention on this matter because most of them use ideal images as the source in order to find pupil location. Ideal orthogonally looking eye images has clear pupil. It is very easy to find the pupil in this case. Unfortunately, the real situations are much complicated.

Although pupil appearance is the same for each user, eye shape is changed when eye moves then pupil appearance is changed accordingly. Also this is a disruption when analyzing pupil for the users who have much eyebrows. Different skin color, race, interference with eyelids, disappearances, and changes of eye shape (when eye move) are the reasons for making the pupil detection very difficult. In this paper, we propose gaze estimation method which is robust against various users as well as illumination changes and also allows user movement. The proposed method utilizes IR camera mounted glass to allow user's movement and to allows to use as a computer input device of a wearable computing. Pupil knowledge such as shape, size, location, and motion are used for eliminating the aforementioned influences. Also the proposed eye-based HCI system allows simultaneous key input of more than three keys, for instance, "Ctl+Alt+Del" for initiating task manager, left/right click and drag/drop of mouse event capabilities. Although commercially available screen keyboard allows simultaneous key input for two keys such as "shit+*", "Alt+*", it does not allow three keys input at once. Such these full specification of mouse events and keyboard functions are available for the proposed eye-based HCI system.

The details of the proposed gaze estimation method with pupil knowledge and eye-based HCI system is described followed by some experiments with six

978-0-7695-4367-3/11 $26.00 © 2011 IEEE
DOI 10.1109/ITNG.2011.81

different nationalities of users. Then some concluding remarks and discussions are followed by in the final section.

2. Proposed method and system

2.1 Proposed gaze estimation method used

In order to analyze eye gaze, eye should be detected and tracked first. Typically, the proposed method detects eye based on deformable template method [27]. This method matches eye template to the previously acquired images. We create eye template by applying Gaussian smoother to the previously acquired image. Deformable template method detects a rough position of eye. Benefit of deformable template method is that it can be done so quickly in comparison to classification methods. Although this method is faster than the classification method, the robustness of the method is still week. In the proposed system, Viola-Jones classifier will detects eye even when deformable template fails to detect eye position. It means that Viola-Jones method is used only when deformable template fails to detect eye. The Viola-Jones classifier employs AdaBoost in a cascade basis to learn a high detection rate. The Viola-Jones classifier function is employed from OpenCV [16]. Before using the function, we should create XML file. The training samples (face or eye image) must be collected. There are two samples: negative and positive sample. Negative sample corresponds to non-object images. Positive sample corresponds to object image. After acquisition of image, OpenCV searches the face center location followed by searching the eye center location. Using combination between deformable eye template and Viola-Jones method, eye location is detected. Advantages of these methods are fast and robust against circumstance changes. After the rough eye position is found, eye image is locked and cropped based on the detected position. It means that we should not repeat the eye detection again. The eye position is not changed because the camera is mounted on the glass. Eye gaze is estimated based on pupil location. Because the method relies on the detected pupil location, it should be done perfectly and accurately and should be robust to a variety of users and circumstance changes. Flow of pupil detection with pupil knowledge is shown in Fig.1. Three types of knowledge are used. We use pupil size, shape, and color as the first knowledge. First, adaptive threshold method is applied to eye image. Threshold value T is obtained from average pixel value (mean) of eye image μ. We set threshold value is 27% bellow from the mean empirically.

$$\mu = \frac{1}{N}\sum_{i=0}^{N-1} I_i \tag{1}$$

$$T = 0.27\mu \tag{2}$$

Pupil is signed as black pixels on image. In the first case, when a pupil clearly appears in eye image, the results of adaptive threshold itself are able to detect pupil location. Pupil is marked as one circle in image. By using connected labeling component method, we can easily estimate the pupil location. Noise or undesired detected object as candidates of pupil can be distinguished with size and shape knowledge as is shown in Fig.2 (a).

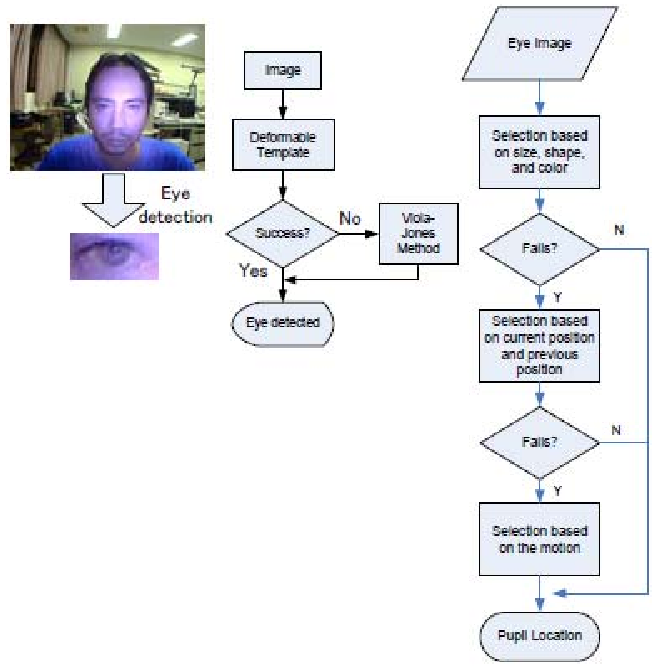
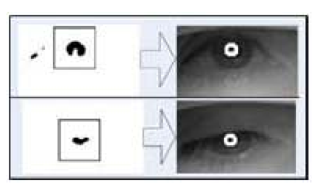
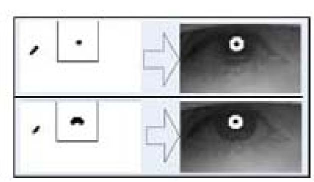


Fig.1 Flow of pupil detection method.

Pupil size and shape is changed in accordance with eye movement when user looks at right or left directions. This condition makes the pupil detection is hard to find. Noise and interference between pupil and eyelid appear. This condition brings through others black pixels which have same size and shape with pupil.



(a) Clearly opened eye



(b) Pupil and noise is almost same size and shape

(c) Pupil is disappear so that no black pixel is extracted

Fig.2 Noise and interference between pupil and eyelid appear.

To solve this problem, we utilize the previously detected pupil location. The pupil is decided using following equation,

$$P(t-1) - C < P(t) < P(t-1) + C \quad (3)$$

The reasonable pupil location $P(t)$ always in the surrounding previous location $P(t-1)$ with the search area C. This case is shown in Fig.2 (b). If all of the aforementioned steps are failed, then we estimate pupil location by its motion. This situation may happen when the black pixels are mixed with the others or no black pixels in the acquired image. We use this knowledge as a last priority of ambiguity motion avoidance between pupil and the other eye components. This case is shown in Fig.2 (c). The figure shows that no black pixels are detected as a pupil in the image (left side).

We track pupil locations using the previously detected location using Kalman filter [18] for prediction of pupil location. Pupil motion at each time instance (frame) can be characterized with position and velocity. Let (i_t, j_t) be the centroid of pupil at time t and (u_t, v_t) be velocity at time t in i and j directions. The state vector at time t can therefore be represented as $x_t = (i_t\, j_t\, u_t\, v_t)^t$. The system can therefore be modeled as,

$$x_{t+1} = \Phi x_t + w_t \quad (4)$$

where wt represents system perturbation. We assume that a fast feature extractor estimates $z_t = (\hat{i}_t, \hat{j}_t)$, the pupil position at time t. Therefore, the measurement model in the form needed by the Kalman filter is,

$$z_t = Hx_t + u_t \quad (5)$$

where u_t denotes measurement uncertainty. We used to storage pupil location of time series history. Given the state model in equation (4) and measurement model in equation (5) as well as some initial conditions, the state vector x_{t+1}, along with its covariance matrix Σ_{t+1}, can be updated using the system model (for prediction) and measurement model (for updating).

The eyeball is assumed to be a sphere with radius R. Actually, it is not quite a sphere but this discrepancy does not affect to the proposed method. The pupil is located at the front surface of eyeball. The distance from the center gaze to current gaze is r. Gaze is defined as angle θ_{eye} between normal gaze and r. The relation between R, r and θ_{eye} is:

$$r = R\sin\theta \quad (6)$$

$$\theta_{eye} = \arcsin\left(\frac{r}{R}\right) \quad (7)$$

The typical eyeball radius ranges from 12 mm to 13 mm according to the anthropometric data [28]. Once r has been found, gaze angle θ_{eye} is calculated easily. Eye gaze is calculated with equation (7) based on r. In order to measure r, the normal gaze is defined. User looks at the center of the screen display first. At this time, we record that this pupil location is normal gaze position. Then the normal gaze position is verified referring to the center of two eye corners (Two ends of the eye). Through this process, drag/drop click events and also troubleshooting becomes available. The existing conventional camera mouse system did drag/drop event by utilize blink. The close eye can be signed as drag event and while open eye is signed as drop event. The weakness of this method is that drag/drop events always rely on eye. Thus it makes users tired and dried their eyes when users use the system for a long duration. The proposed method overcomes such problem.

The modified screen keyboard layout is shown in Fig.3. We added two buttons, task manager and drag/drop. Although the existing screen keyboard system allows input two keys, such as “Sft+*”, Alt+*”, it does not allow input more than three keys such as “Ctl+Alt+Del” so that trouble shooting cannot be done for the existing screen keyboard. When user uses the camera mouse and not responding program happens, user can click this button to end task the not-responding program. The second button is drag/drop button which assist user to drag and drop the object. For instance, when user want to move the window position of running program, user can click this drag/drop button and continue with click to the window and then the last click will be automatically signed as drop event.

Fig.3 Modified Screen Keyboard

Thus the proposed mouse assist user does interaction with computer by click the button or other graphical user interface tool. The typical mouse usually has two buttons: (1) left click button (performs a first action on object selection) and (2) right click button (performs a secondary action). After mouse cursor is moved, the decision is done by these click events. For instance, when mouse is used for typing using screen keyboard software, user select the desired key by click the key. Typically this clicking is done by using left click. Also, when user wants to show the properties menu of icon, user does right click on this icon. Both of left and right clicks can be used simultaneously results in user become easy to interact with computer. In this paper, we propose mouse click events using timer (fixed one’s eye on for a while) and blink. The proposed system uses timer as left click event. By using timer, left click will automatically click when cursor position is not moved during specific period of time. This method is easier than the other method such as blinking, head rotation, opened mouth. It causes no physiological impact. Typically left click event is more frequently used than right click. We cannot use blinking method as left click because the uses of blinking for a long period of time that may cause the user's eye become dry. We use blinking as right click event. Typical accidental blink is done for 0.3 second so that intentional blink is defined as eye is closed for more than 0.4second (1 second in the proposed method).

The both click events run simultaneously. In left click event, system finds the steady of cursor position. If the steady of cursor position is less than a previously determined threshold, timer 1 is activated while the steady of cursor is more than threshold, timer 1 is reset. If timer 1 is continued for more than 1 second, system send left click event while timer 1 is turned off. The steady of cursor position is calculated with the following equation (8),(9),

$$\mu(x,y) = \sum_{i,j} \frac{P(x_i, y_j)}{N}, \tag{8}$$

$$Steady = \sum_{N} |P_N(x) - \mu_N(x)| + |P_N(y) - \mu_N(y)|, \tag{9}$$

where P(x, y) is cursor location and N is the number of locations. The left click event can be done by using timer. Blink is detected by using motion template method [16]. This method obtains magnitude and angle of total motion in the acquired image. After eye image is cropped, motion template method estimates the motion. We detect blink if magnitude of motion in the image is high and the first angle is 270º (eye become closed) and the opened eye is signed by angle is 90º. First, we detect the closed eye by using motion template method. When eye become closed, timer is turned on. If eye becomes open when timer 2 is set bellow 1 second, it is not recognized as blink. Otherwise, if eye becomes open and timer 2 is set over second, it is recognized as blink. If blink happens, system sends right click event.

3. Experiment and Results

3.1 System configuration

The proposed system utilizes only one single IR web camera NetCowBoy DC-NCR 131 to acquire users' face image. To eliminate influences due to illumination changes, the IR camera which includes IR light emitting diodes is used. User looks at the screen while camera put on top of screen display. Only camera that mounted on top of screen display is required. Optiplex 755 dell computer with Core 2 Quad 2.66 GHz CPU and 2G RAM is used. We develop our software under C++ Visual Studio 2005 and OpenCv Image processing Library which can be downloaded as free on their website. The specification of IR camera is shown in Table 1. The pointing value is determined based on head poses and eye gaze.

Face and eye is detected with AdaBoost based on Haarlike feature method (proposed by Viola and Jones) [29]. Then eye pupil is detected with knowledge of the typical pupil. Eye gaze direction is defined as the line of sight which connects eye-ball center (typical eye-ball size is assumed to be 24mm) and pupil center detected already. Once eye gaze is detected, then template matching with adaptive threshold is followed. Matching point of the line of sight on the computer screen is cursor position. In order to update the cursor position, Lucas-Kanade optical flow tracks the position. Several image processing methods is used to obtain cursor coordinate. Cursor coordinate is determined based on head position and eyes gaze itself. Blink detection is used to make selection command.

Table 1 Specification of IR Camera

Pixel	1.3 Million
Resolution	1280×1024
Frame rate	1280 x 1024：7.5fps, 640 x 480：30fps
Dimension	52mm (W) × 65mm (D) × 70mm (H)
Weight	85g
Operating condition	0 - 40deg.C
Interface	USB 2.0
IR Illumination	7 IR LED

3.2 Eye gaze estimation experiments

Eye gaze estimation experiments include eye detection success rate with different users, illumination influence, noise influence, and gaze estimation accuracy. The eye detection experiment is carried out with six different users who have different race and nationality: Indonesian, Japanese, Sri Lankan, and Vietnamese. We collect data from each user while making several eye movements. Three of Indonesian eyes who have different race are collected. The collected data contain several eye movement such as look at forward, right, left, down, and up. Two of Indonesian eyes have width eye with clear pupil. The numbers of images for them are 552 and 668 samples, respectively. Another Indonesian eye has slanted eyes and the pupil is not so clear. The number of images of this user is 882 samples. We also collected data from Sri Lankan people. His skin color is black with thick eyelid. The number of images is 828 samples. The collected data of Japanese is bright with slanted eyes. The number of images is 665 samples.

The first experiment investigates the pupil detection accuracy and variance against various users. We count the success samples followed by counting the success rate. The proposed method (adaptive threshold method with pupil knowledge) is compared to the adaptive threshold method and Template matching method. The adaptive threshold method uses combination between adaptive thresholds itself and connected labeling method. The template matching method use pupil template as reference and matched with the images. The experimental result is shown in Table 2. The result shows that the proposed method has much higher success rate than the others. Also, the proposed method is robust against the various users (the variance is 16.27). Influence of illumination changes on eye detection success rate has been clarified. Adjustable light source with multi-functional environmental detector LM-8000 for monitoring the illumination is used in the experiment. Experimental result shows that the proposed method works with zero illumination condition (dark place) because IR camera has IR LED. The proposed method does not work for very bright illumination condition.

The next experiment measures noise influence against eye detection success rate. Normal distribution of random number generator derived noise is added to the acquired images. Normal distribution with 0 mean and the designated standard deviation is assumed. The experiment data show that our system robust enough due to noise influence.

In more detail, effectiveness of knowledge on pupil detection success rate is evaluated.

Table 2. Robustness against various users, this table shows that the proposed method robust enough against varies user and also has high success rate.

No	Nationality	Our Method (%)	Adaptive Threshold Method (%)	Template Matching Method (%)
1	Indonesian	97.05	97.05	7.18
2	Indonesian	95.44	78.83	56.57
3	Indonesian	78.83	77.72	58.22
4	Vietnamese	95.50	95.16	38.93
5	Srilanka	87.42	74.12	47.88
6	Indonesian	97.00	95.63	64.17
7	Japanese	80.55	78.25	51.00
	Average	90.26	85.25	46.28
	Variance	63.15	102.66	362.65

Table.3 Effectiveness of each pupil knowledge method step

No	Nationality	Step 1 (%)	Step 2 (%)	Step 3 (%)
1	Indonesian	56.91	25.87	14.27
2	Indonesian	60.48	18.35	16.61
3	Indonesian	12.53	66.02	0.28
4	Vietnamese	95.16	0.00	0.35
5	Srilanka	71.70	2.42	13.30
6	Indonesian	86.39	10.36	0.25
7	Japanese	51.45	26.95	2.14
	Average	62.09	21.42	6.74

The results show that the most effective knowledge is pupil size shape and color (Step 1) followed by the previously detected pupil location (Step 2) and the pupil motion (Step 3) as is shown in Table 3.
Eye gaze estimation accuracy is evaluated. Users look at the several points and then measure the distance between the designated position and the estimated position.
The error is calculated from the designated angle and the actual angle. The errors are shown in Table 4. The experimental results show that the proposed method is accurate enough for eye-mouse (the accuracy is 0.64^{o}).

Table 4 Eye gaze accuracy (Average error=0.64)

Point	Error (^{0})
1	0
2	0.2
3	0.2
4	0.2
5	3.12

In order to determine the selected key with the eyemouse, (1) Fix one's eyes on (timer mode) or (2) Blink (blinking mode) is used. User can designate one of those. In the timer mode, the selected key is determined when the mouse cursor location changes within the range of stability for 500 ms. Meanwhile, the selected key is determined when closed eye is detected for more than 500 ms (Accidental blink is occurred within 300 ms typically). For the blinking mode, determination of selected key is performed when eye status is changed as open-close-open state so that blinking mode takes much longer time compared to the timer mode. Experimental results show that the success rate for the timer mode is 100% and it is proportional to time because it does not give any influence to users.

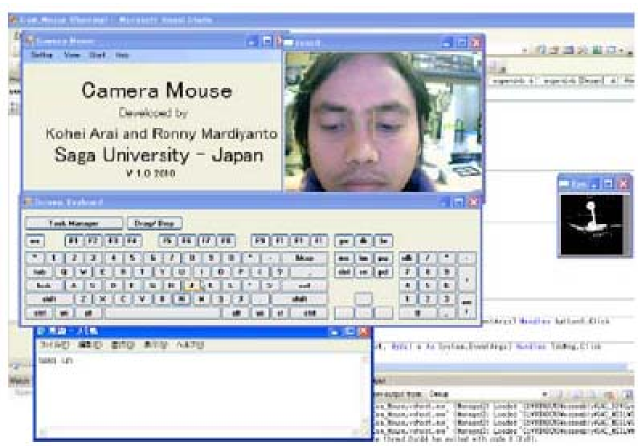


Fig.4 Example of the computer screen

Although the success rate for the blinking mode is also 100%, it is only for a first one hour. Blink makes users

tired when user uses blinks for a long period of time, then it may cause fatigue on their eyes results in degradation of success rate in accordance with time duration. In order to measure success rate, IR camera is set-up on the top of the computer screen. The computer screen size is 41 cm by 25.5 cm. The set-up situation is shown in Fig4.
In the middle of the figure, modified screen keyboard is situated. There are two additional keys of “task manager” and “drag/drop”. At the top middle, acquired user’s face is displayed together with yellow square mark of eye and surroundings. At the right middle, there is binalized image of eye and surroundings with estimated gaze direction. At the left bottom, there is Microsoft Wardpad window in which selected and determined keys or sentences are displayed. This demonstration can be seen at http://www.youtube.com/watch?v=tKFs87wYVKA.

4. Concluding Remarks

Eye-based HCI method and system using pupil knowledge has been successfully implemented. It shows good enough success rates of more than 96% for key and mouse event control and selection and determination without any prior calibration process. Also it is robust to various users, illumination changes, and head movement. The proposed method solved the problem that the existing conventional eye-mouse and camera-mouse is used to fail to detect pupil location. The proposed method uses the following priority of knowledge in order to detect the pupil. (1) Size, shape, and color as a first priority of knowledge, followed by previously detected pupil (2) location and (3) motion. It is concluded that the proposed method achieved 5% of improvement on gaze estimation accuracy through the comparison between with and without using the knowledge in pupil detection process.

References

[1] Park, K.S., Lee, K.T.: Eye controlled human/computer interface using the line of sight and intentional blink. Computer Ind. Eng. 30(3), 463–473 (1996)
[2] Ito, Sudo, Ifuku: The look input type communication equipment for serious physically handicapped persons. The Institute of Electronics, Information and Communication Engineers paper magazine J83D1C5, 495–503 (2000)
[3] Yamada, F.: The text creation and the peripheral equipment control device by eye movement. The Institute of Electronics, Information and Communication Engineers paper magazine CJ69D(7), 1103–1107 (1986)
[4] Yamada, M.: The research trend of the latest eye movement, An Electric Information-and- Telecommunications Academic Journal, MBE95-132, NC95- 90,145-152 (1995)
[5] Abe, K., Ohiamd, S., Ohyama, M.: An Eye-gaze Input System based on the Limbus Tracking Method by Image Analysis for Seriously Physically Handicapped People. In: 7th ERCIMWorkshop User Interface for All Adjunct Proc., pp. 185–186 (2002)
[6] Creative Advanced Technologies, http://www.creact.co.jp/jpn/por.pdf
[7] Kuno, Y., Fujii, K., Uchikawa: Development of the look input interface using EOG. The Information Processing Society of Japan Paper Magazine C39C5, 1455–1462 (1998)
[8] Robinson, D.A.: A method of measuring eye movement using a sclera search coil in a magnetic field. IEEE Trans. on Biomedical Electronics 10, 137–145 (1963)
[9] The organization of the retina and visual system, http://webvision.med.utah.edu/
[10] Ito, N.: Eye movement measurement by picture taking in and processing via a video capture card. An Institute of Electronics, Information and Communication Engineers Technical Report C102 C128 C31-36 (2002)
[11] Kishimoto, Y., Hirose, C.: Development of the look input system by a cursor move system. The Institute of Image Information and Television Engineers C55C6C917-919 (2001)
[12] Corno, L., Farinetti, I.,, S.: A Cost-Effective Solution for Eye-Gaze Assistive Technology. In: Proc. IEEE International Conf. on Multimedia and Expo., vol. 2, pp.433–436 (2002)
[13] Abe, O., Oi, D.: The look input system using the sclera reflection method by image analysis. The Institute of Image Information and Television Engineers 57(10), 1354–1360 (2003)
[14] Abe, Daisen, Oi.: The multi-index look input system which used the image analysis under available light. The Institute of Image Information and Television Engineers, 58C11C 1656-1664 (2004)
[15] Abe, Daisen, Oi.: The look input platform for serious physically handicapped persons, Human Interface Society Human interface symposium 2004 collected papers C1145-1148 (2004)
[16] Bradski, G., Kaebler, A.: Learning Computer Vision with the OpenCV Library, pp. 214–219. O’Reilly, Sebastopol (2008)
[17] Haro, A., Flickner, M., Essa, I.: Detecting and Tracking Eyes By Using Their Physiological Properties, Dynamics, and Appearance. In: Proceeding of CVPR 2000, pp. 163–168 (2000)
[18] Zhu, Z., Ji, Q., Fujimura, K., Lee, K.: Combining Kalman filtering and mean shift for real time eye tracking under active IR illumination. In: Proceeding of 16th Pattern Recognition International Conference, vol. 4, pp. 318–321 (2002)
[19] Takegami, T., Gotoh, T., Kagei, S., Minamikawa-Tachino, R.: A Hough Based Eye Direction Detection Algorithm without On-site Calibration. In: Proceeding of 7th Digital Image Computing: Techniques and Applications, pp. 459–468 (2003)
[20] Lam, K.M., Yan, H.: Locating and extracting eye in human face images. Pattern Recognition 29(5), 771–779 (1996)
[21] Chow, G., Li, X.: Towards a system of automatic facial feature detection. Pattern Recognition (26), 1739– 1755 (1993)
[22] Rajpathaka, T., Kumarb, R., Schwartzb, E.: Eye Detection Using Morphological and Color Image Processing. In: Proceeding of Florida Conference on Recent Advances in Robotics (2009)
[23] Brunelli, R., Poggio, T.: Face Recognition: Features versus templates. IEEE Trans. Patt. Anal. Mach. Intell. 15(10), 1042–1052 (1993)
[24] Beymer, D.J.: Face Recognition under varying pose. In: Beymer, D.J. (ed.) IEEE Proceedings of Int. Conference on Computer Vision and Pattern Recognition (CVPR 1994), Seattle, Washington, pp. 756–761 (1994)
[25] Shafi, M., Chung, P.W.H.: A Hybrid Method for Eyes Detection in Facial Images. International Journal of Electrical, Computer, and Systems Engineering, 231–236 (2009)
[26] Morimoto, C., Koons, D., Amir, A., Flickner, M.: Pupil detection and tracking using multiple light sources. Image and Vision Computing 18(4), 331–335 (2000)
[27] Yuille, A., Haallinan, P., Cohen, D.S.: Feature extraction from faces using deformable templates. In: Proceeding of IEEE Computer Vision and Pattern Recognition, pp. 104–109 (1989)
[28] Wang, J.-G., Sung, E., Ronda Venkateswarlu.: Eye gaze estimation from a single image of one eye, Proceeding of Ninth IEEE International Conference on Computer Vision, vol.1, pp. 136-143, (2003)
[29] Viola, Jones.: Robust real-time object detection , (2001)