TELKOMNIKA, Vol.9, No.2, August 2011, pp. 377~386
ISSN: 1693-6930
accredited by DGHE (DIKTI), Decree No: 51/Dikti/Kep/2010

Low-Cost Based Eye Tracking and Eye Gaze Estimation

I Ketut Gede Darma Putra*[1], Agung Cahyawan[2], Yandi Perdana[3]
Departement of information technology and Electrical Engineering, Udayana University
Bukit Jimbaran, Bali, Indonesia, phone +62361703315
e-mail: duglaire@yahoo.com*[1], agungcahyawan@ee.unud.ac.id[2], yandipradana@yahoo.com[3]

Abstrak

Harga dari sistem penjejak gaze untuk penggunaan masyarakat umum masih sangat mahal. Alasan utama mahalnya sistem penjejak gaze adalah karena kamera dan lensa yang digunakan untuk membuatnya berkualitas tinggi dan tingginya biaya pengembangan. Penelitian ini membangun sistem penjejak gaze berbiaya rendah. Alat dibuat dengan melakukan modifikasi pada kamera web agar bekerja dalam spektrum infrared. Teknik baru juga diperkenalkan disini untuk mendeteksi koordinat pusat pupil menggunakan connected component labeling. Dengan mengkombinasikan teknik tersebut dengan regresi polinomial orde 3 dalam proses kalibrasi untuk menentukan titik gaze. Hasil pengujian menunjukkan sistem ini mampu memberikan akurasi yang dapat diterima dengan galat piksesl 0.39^o dalam derajat visual.

Kata kunci: *connected component labeling, penjejak gaze, infrared, kamera web, regresi polinomial*

Abstract

The costs of current gaze tracking systems remain too high for general public use. The main reason for this is the cost of parts, especially high-quality cameras and lenses and cost development. This research build the low cost based for gaze tracking system. The device is built by utilizing of modified web camera in infrared spectrum. A new technique is also proposed here in order to detect the center pupil coordinate based on connected component labeling. By combination the pupils coordinate detection method with third order polynomial regression in calibration process to determine the gaze point. The experiment results show our system has an acceptable accuracy rate with error pixel 0.39^o in visual degree.

Keywords: *webcam, infrared, gaze tracking, connected component labeling, polynomial regression.*

1. Introduction

Gaze tracking has been used for many decades as a tool to study human cognitive process including reading, driving, watching commercials, and other activities [1]. With the advent of personal computers, potential integration of such systems has been considered only recently in 1991 [2]. Successful attempts to employ gaze tracking as user interface were made to allow users with movement disabilities to type by looking at virtual keyboard [3], or for mouse pointer control [4]. Systems of this kind have also been used by individuals without any disabilities to enhance performance [5],[6]. In recent years, gaze tracking is widely used in the areas of intelligent control [7], virtual reality, video games, robotics, human computer interaction, eye diseases diagnosis, human behavior studies, etc [8]. Lee et al used gaze tracking system for controlling IPTV [9].

Gaze Tracking system detect the eye location in image and estimate the gaze path by measurement of point of gaze (POG). The eye position is commonly measured using the pupil or iris center. The detected eyes in the images are used to estimate and track where a person is looking in 3D or alternatively determining the 3D line of sight [10]. Gaze tracking system is divided into head mounted or wearable-camera-based or intrusive system which required direct contact with the eye, as in [11],[12] and head-free or remote based or nonintrusive system which avoid any physical contact with the user as in [13],[14],[15].

The continuing absence of consumer-grade eye-tracking human computer interface technology for general public use is result of the high price (cost) of eye tracking technology, and intrusiveness of such systems, despite the fact that technologies allowing so have existed for many years [16]. The main reason for this is the cost of parts, especially high-quality

Received March 28th, 2011; Revised June 12th, 2011; Accepted June 20th, 2011

cameras and lenses, the cost of development, and the relatively small market [10]. Only recently relatively low cost systems have been investigated as in [17],[18].

The design work described herein shows the accuracy that can be obtained using modern, low-cost systems. The three most significant contributions of this research are the design of hardware, in this case a webcam [19] for worked in the infrared spectrum, use of blob/connected component labeling algorithm to detect the pupil in the infrared spectrum instead of iris detection in the visible spectrum, and use of higher dimensional polynomials to calibrate and map the detected gaze position to the position on the computer screen.

In a gaze tracking system, images of the eye are taken by a camera and sent to an image processing system. This image data is a picture of the user's eye from a specific vantage point. The vantage point can be close to the user's eye, as in a head mounted device or further away, as in a device on a table near the user. For this project, the design team has used a head mounted camera, as will be discussed later. The image is processed to determine the location of the user's pupil within the image. The coordinates of the detected center of the user's pupil are passed through a specialized set of algorithms to accurately position a cursor on a screen displayed in front of the user. The cursor position on the screen represents where the user is looking on the display. This creates an open loop tracking system where a cursor follows where the user is looking on the screen.

2. Research Method

2.1. General System Overview

In general the system is built in this research consist of two main modules as illustrated in Figure 1. The first module is hardware design, which is modification of the webcam in order to have worked in infrared spectrum in such way so that the pupil relative easier to be detected and tracked.

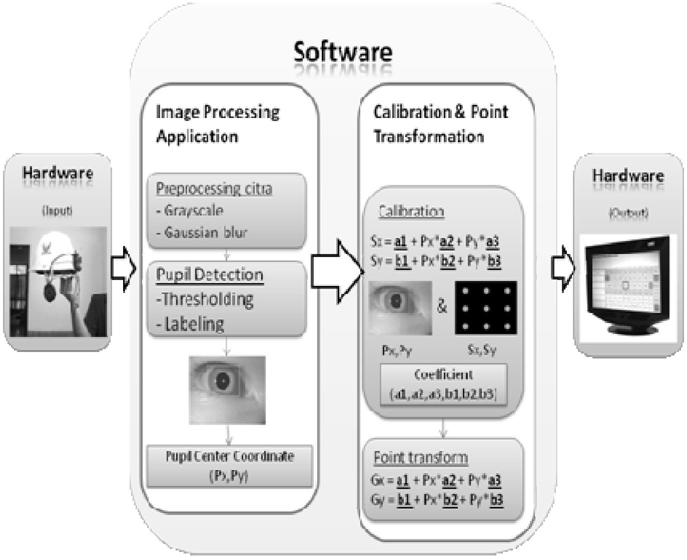


Figure 1. General overview of the system

The second module is software design, which is the main purpose are for detect and track the pupil coordinate using thresholding and connected component labeling algorithm, transform that coordinate to monitor coordinate through calibration and point transformation process using polinomial regression, and the last is testing process to measure accuration rate in term of error pixel between gaze coordinate result from calaculation of the system and aimed monitor coordinate.

2.2. Hardware Design

The hardware design of this system consist of 4 main process :

1. Infrared filter removal

Every webcam in general equipped with infrared filter for blocking the infrared light and allowing visible light. Because in this research will use infrared spectrum approach, so that the webcam must be modified to remove the infrared filter as shown in Figure 2. Infrared light is useful for eye trackers, mainly because it is not only invisible to the user but also it can be used for controlling light conditions, obtaining higher contrast images, and stabilizing gaze estimation [8].

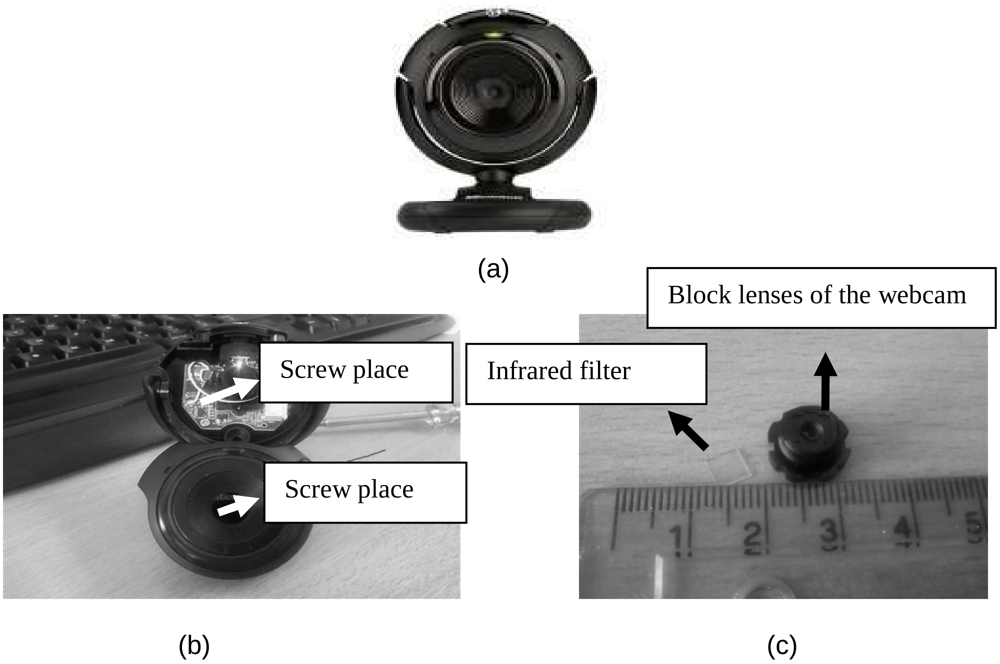


Figure 2. Infrared filter removal, (a) Webcam Microsoft Lifecam VX-1000, (b) Webcam after the screw is closed, (c) Infrared filter is removed

2. Attachment of visible light filter

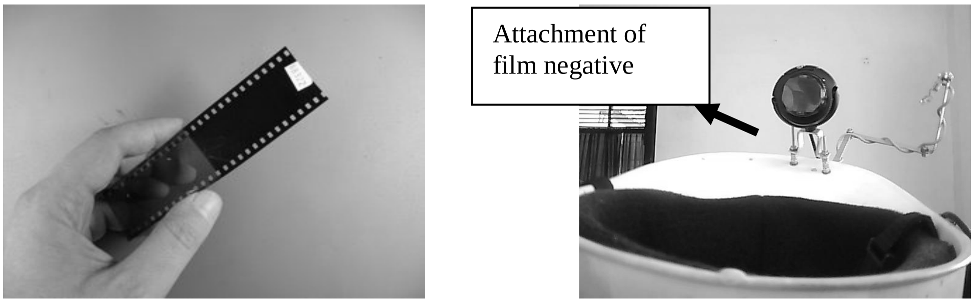
The next process is to attach a filter to block visible light and let only infrared light into the webcam as shown in Figure 3. The filter is film negative already to be printed and take the black part of that film, usually in the beginning and the end of roll film. Price of the original filter to block the visible light is very expensive, so film negative is used instead, because the main purposes of this research is to built eye tracking and eye gaze tracking system based on low cost component.

3. Making of infrared light source

After the webcam has been modified, the next step is supplies the infrared light source for the webcam. In this research infrared light source is built using 1 infrared LED which connected into 2 battery each of them 1.5 volt, resistor 33 ohm, and 1 switch to turn on or off the infrared LED. This research use only 1 LED after consider that webcam will be positioned very close with the eye, so 1 LED is enough for infrared source.

4. Finalizing

The last step to design hardware is linking up together all the components in a helmet as shown in Figure 5. This helmet later will have weared by the users of the system.



(a) (b)

Figure 3. Attachment of visible light filter, (a) film negative, (b) Attachment of film negative to block visible light

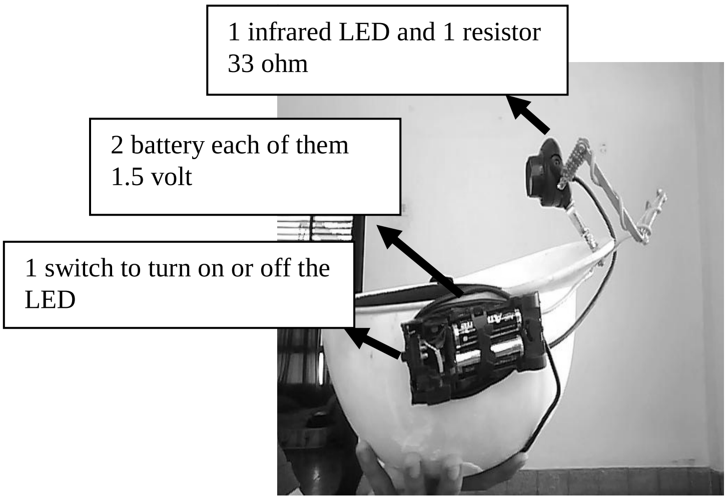
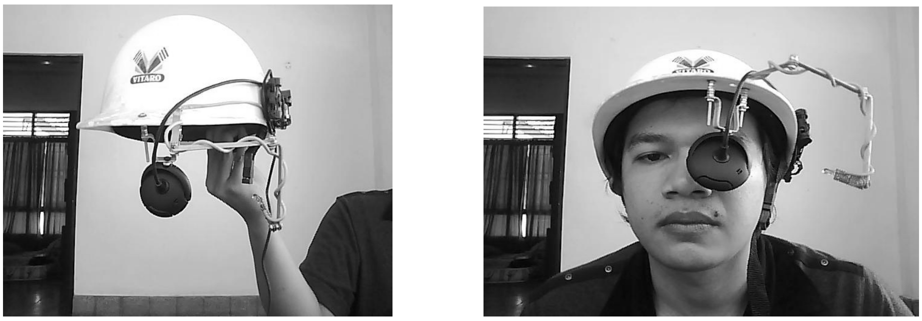

1 infrared LED and 1 resistor 33 ohm
2 battery each of them 1.5 volt
1 switch to turn on or off the LED

Figure 4. Infrared LED, resistor, switch and batteries



(a) (b)

Figure 5. (a) Final hardware, (b) hardware weared by user

2.3. Software Design

There are two main subsystems to the software. The first is the image processing application, which processes the image and locates the center of the pupil. The second is the calibration and point transformation subsystems, which do the mapping from the center coordinate of the eye pupil to the screen coordinate. The software is developed by using Visual

C++ and OpenCV library. The intricacies of these subsystems will be described in more detail in the following sections.

2.3.1. Detection of the center coordinate of the pupil

The pupil may be darker than their surroundings and thresholds may be applied if the contrast is sufficiently large. Yang et al and Stiefilhagen at al introduce an iterative threshold algorithm to locate the pupils by looking for two dark regions that satisfy certain anthropometric constraints using a skin-color model. Their method is limited by the results of the skin-color model and it will fail in the presence of other dark regions such as eyebrows and shadows [10]. Yang et al applied the ellipse fitting algorithm to fit a standard ellipse or circle based on the coordinates of pupil edge pixels. The center of the ellipse or circle is the center of the pupil [7]. This research use simple technique to detect the pupil center using connected components labeling.

The steps to detect the center of pupil is used here are shown in Figure 6. The process begins by capturing a eye image frame from the camera via OpenCV's camera subsystem. The grayscaling process convert the RGB color to gray color space. After that the grayscale image is smoothed using Gaussian filter to remove any noise in image. The smoothing also helps to reduce sharp edges, aiding the pupil detection system.

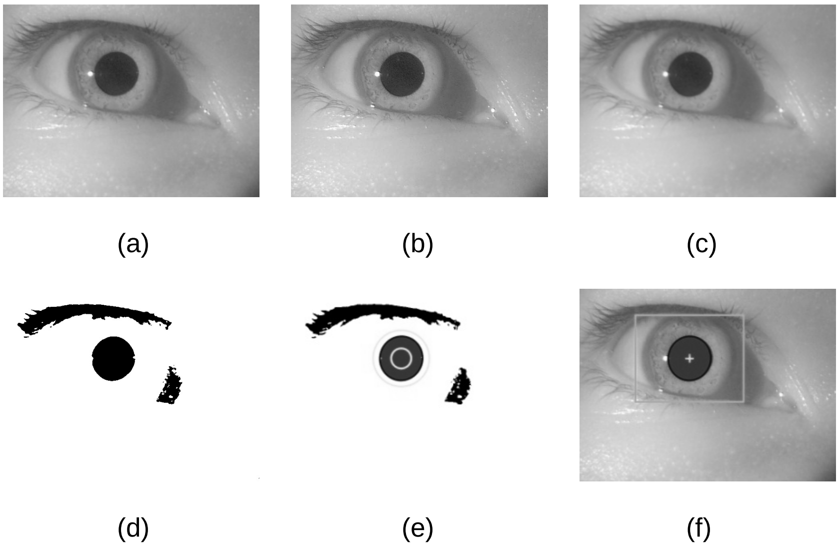


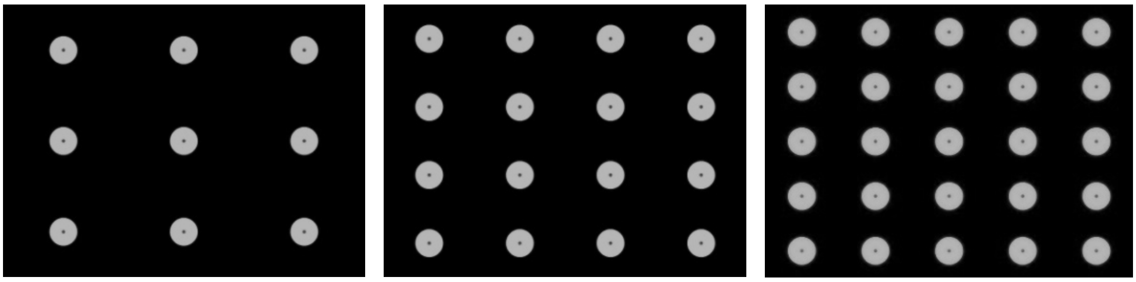
Figure 6. Detection of the center of pupil, (a) capture frame from camera, (b) grayscale image, (c) Gaussian blur image, (d) binary image, (e) component labelling image, (f) output image

The next step is thresholding the smoothed image to obtain the binary image. The pupil will be clearly visible as a black component in this image. However, there may be other black component as well. These represent areas of the image of almost exactly the same shade and color as the pupil. The connected component labeling step exists here to locate all black components in the image and determine which is representative of the pupil. A connected component is defined as a group of pixels with values within a certain range. The connected component labeling algorithm used is an open source add-on for OpenCV.

Once all connected component have been located, the system calculates several parameters of each blob such as area, aspect ratio, roundness, and more. These parameters are compared to experimentally determined values for a pupil and connected componentes are discarded based on them. Due to the nature of the human eye and surrounding features, there will never be more than one connected component that fits all parameters for a human eye. Thus, the system selects the correct connected component. The algorithm then calculates the center coordinate of this connected component.

2.3.2. Calibration and Point Transformation

Calibration is necessary due to the fact that a screen monitor is a flat n by m pixel rectangle while the human eye is not. Mapping is required to transform the coordinates of the center of the pupil to the coordinates on the display. Calibration must be done every time the system is restarted due to variations in use. The eye will not be in the same location relative to the screen every time the same user wears it, also, different users with different eye and face shapes will require a new calibration. In this research, the calibration process is done using 9, 16, or 25 pixel locations, by asking the user to 'look at the dot' in monitor. The pixel locations can be shown in Figure 7.



(a) (b) (c)

Figure 7. Calibration pixel locations, (a) 9 pixel, (b) 16 pixel, (c) 25 pixel

By using the center location of the eye when looking at those known pixel locations the coefficients of the calibration equation are determined using polynomial regression similar with [8], [20], [21]. This paper used first (equation (1) and (2)), second (equation (3) and (4)), or third (equation (5) and (6)) order polynomial regression. Polynomial regression is a statistical technique used to approximate correlation of variables. Shown below the polynomial regression formula:

$$Sx = KoefX1 + Px * KoefX2 + Py * KoefX3 \tag{1}$$

$$Sy = KoefY1 + Px * KoefY2 + Py * KoefY3 \tag{2}$$

$$\begin{aligned} Sx = {} & KoefX1 + Px * KoefX2 + Py * KoefX3 + Px^2 * \\ & KoefX4 + Px * Py * KoefX5 + Py^2 * KoefX6 \end{aligned} \tag{3}$$

$$\begin{aligned} Sy = {} & KoefY1 + Px * KoefY2 + Py * KoefY3 + Px^2 * \\ & KoefY4 + Px * Py * KoefY5 + Py^2 * KoefY6 \end{aligned} \tag{4}$$

$$\begin{aligned} Sx = {} & KoefX1 + Px * KoefX2 + Py * KoefX3 + Px^2 * \\ & KoefX4 + Px * Py * KoefX5 + Py^2 * KoefX6 + Px^3 * \\ & KoefX7 * Py * KoefX8 + Px * Py^2 * KoefX9 + Py^3 * KoefX10 \end{aligned} \tag{5}$$

$$\begin{aligned} Sy = {} & KoefY1 + Px * KoefY2 + Py * KoefY3 + Px^2 * \\ & KoefY4 + Px * Py * KoefY5 + Py^2 * KoefY6 + Px^3 * \\ & KoefY7 * Py * KoefY8 + Px * Py^2 * KoefY9 + Py^3 * KoefY10 \end{aligned} \tag{6}$$

(Px, Py) and (Sx, Sy) represents center pupil coordinate and target (screen monitor) coordinate respectively. With 9,16, or 25 points sample/training points in the monitor will produce 9,16, or 25 equations for each of this (1), (2), (3), (4), 5) and (6) equations. To obtain regression coefficients is used least square method. This method will convert the equations in matrix form (7) and (8) (here only shown sample for first order polynomial from equation 1 and 2).

$$\begin{bmatrix} \mathrm{n} & \sum_{i=1}^{n} \mathrm{Px_i} & \sum_{i=1}^{n} \mathrm{Py_i} \\ \sum_{i=1}^{n} \mathrm{Px_i} & \sum_{i=1}^{n} \mathrm{Px^2{}_i} & \sum_{i=1}^{n} \mathrm{Px_i\,Py_i} \\ \sum_{i=1}^{n} \mathrm{Py_i} & \sum_{i=1}^{n} \mathrm{Px_i\,Py_i} & \sum_{i=1}^{n} \mathrm{Py^2{}_i} \end{bmatrix} \begin{bmatrix} \mathrm{KoefX1} \\ \mathrm{KoefX2} \\ \mathrm{KoefX3} \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^{n} \mathrm{Sx_i} \\ \sum_{i=1}^{n} \mathrm{Px_i\,Sx_i} \\ \sum_{i=1}^{n} \mathrm{Py_i\,Sx_i} \end{bmatrix} \quad (7)$$

$$\begin{bmatrix} \mathrm{n} & \sum_{i=1}^{n} \mathrm{Px_i} & \sum_{i=1}^{n} \mathrm{Py_i} \\ \sum_{i=1}^{n} \mathrm{Px_i} & \sum_{i=1}^{n} \mathrm{Px^2{}_i} & \sum_{i=1}^{n} \mathrm{Px_i\,Py_i} \\ \sum_{i=1}^{n} \mathrm{Py_i} & \sum_{i=1}^{n} \mathrm{Px_i\,Py_i} & \sum_{i=1}^{n} \mathrm{Py^2{}_i} \end{bmatrix} \begin{bmatrix} \mathrm{KoefY1} \\ \mathrm{KoefY2} \\ \mathrm{KoefY3} \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^{n} \mathrm{Sy_i} \\ \sum_{i=1}^{n} \mathrm{Px_i\,Sy_i} \\ \sum_{i=1}^{n} \mathrm{Py_i\,Sy_i} \end{bmatrix} \quad (8)$$

KoefY1, KoefY2, KoefY3 are the regression coefficients that will be computed. The regression coefficients are calculated using Gauss Elimination methods. The next step after the coefficients are obtained is transform or mapping the pupil coordinate to screen coordinate to get the gaze point *(Gx,Gy)*. This point can be achieved by multiplication of pupil coordinate output from detection of center pupil with regression coefficients output from calibration process. The first (equation (9) and (10)), second (equation (11) and (12)), and third (equation (13) and (14)) order polynomial gaze point can be computed as below:

$$\mathrm{Gx} = \mathrm{KoefX}[0] + \mathrm{Px} * \mathrm{KoefX}[1] + \mathrm{Py} * \mathrm{KoefX}[2] \quad (9)$$

$$\mathrm{Gy} = \mathrm{KoefY}[0] + \mathrm{Px} * \mathrm{KoefY}[1] + \mathrm{Py} * \mathrm{KoefY}[2] \quad (10)$$

$$Gx = KoefX[0] + Px * KoefX[1] + Py * KoefX[2] + Px^2 * KoefX[3] + \quad (11)$$

$$\begin{aligned} \mathrm{Gy} = {} & \mathrm{KoefY}[0] + \mathrm{Px} * \mathrm{KoefY}[1] + \mathrm{Py} * \mathrm{KoefY}[2] + \mathrm{Px}^2 * \mathrm{KoefY}[3] + \\ & \mathrm{Px} * \mathrm{Py} * \mathrm{KoefY}[4] + \mathrm{Py}^2 * \mathrm{KoefY}[5] \end{aligned} \quad (12)$$

$$\begin{aligned} \mathrm{Gx} = {} & \mathrm{KoefX}[0] + \mathrm{Px} * \mathrm{KoefX}[1] + \mathrm{Py} * \mathrm{KoefX}[2] + \mathrm{Px}^2 * \mathrm{KoefX}[3] + \\ & \mathrm{Px} * \mathrm{Py} * \mathrm{KoefX}[4] + \mathrm{Py}^2 * \mathrm{KoefX}[5] + \mathrm{Px}^3 * \mathrm{KoefX}[6] + \\ & \mathrm{Px}^2 * \mathrm{Py} * \mathrm{KoefX}[7] + \mathrm{Px} * \mathrm{Py}^2 * \mathrm{KoefX}[8] + \mathrm{Py}^3 * \mathrm{KoefX}[9] \end{aligned} \quad (13)$$

$$\begin{aligned} \mathrm{Gy} = {} & \mathrm{KoefY}[0] + \mathrm{Px} * \mathrm{KoefY}[1] + \mathrm{Py} * \mathrm{KoefY}[2] + \mathrm{Px}^2 * \mathrm{KoefY}[3] + \\ & \mathrm{Px} * \mathrm{Py} * \mathrm{KoefY}[4] + \mathrm{Py}^2 * \mathrm{KoefY}[5] + \mathrm{Px}^3 * \mathrm{KoefY}[6] + \\ & \mathrm{Px}^2 * \mathrm{Py} * \mathrm{KoefY}[7] + \mathrm{Px} * \mathrm{Py}^2 * \mathrm{KoefY}[8] + \mathrm{Py}^3 * \mathrm{KoefY}[9] \end{aligned} \quad (14)$$

This gaze point can be used to control the movement of mouse cursor using our eye.

3. Results and Analysis

To obtain performance of this system, the testing used data from 10 users. Each user do 9 model testing (combination of 9, 16, 25 sampel point with first, second and third order polynomial). From each model testing of each user is calculated the gaze points and calculate the distance (error pixels) between the point and 36 testing points in the monitor screen using euclidean distance [22] (see Figure 8). Average (avgError) and maximal (maxError) error distance from 36 distances from each model testing of each user are computed. To measure which model has the best performance, four indicators below are computed using 10 users in offline mode:

- average distance/error pixel (finalAvgError)
- average maximal distance/error pixel (finalMaxError)
- standard deviation between avgError and finalAvgError (finalStdDeviasiAvgError)
- standard deviation between MaxError and finalMaxError (finalStdDeviasiMaxError).

Gray circles in Figure 8 represents target points, while black cross sign represents gaze point obtained from testing. The Figure is obtained from model 9 which combine 25 sample points with third order regression polynomial.

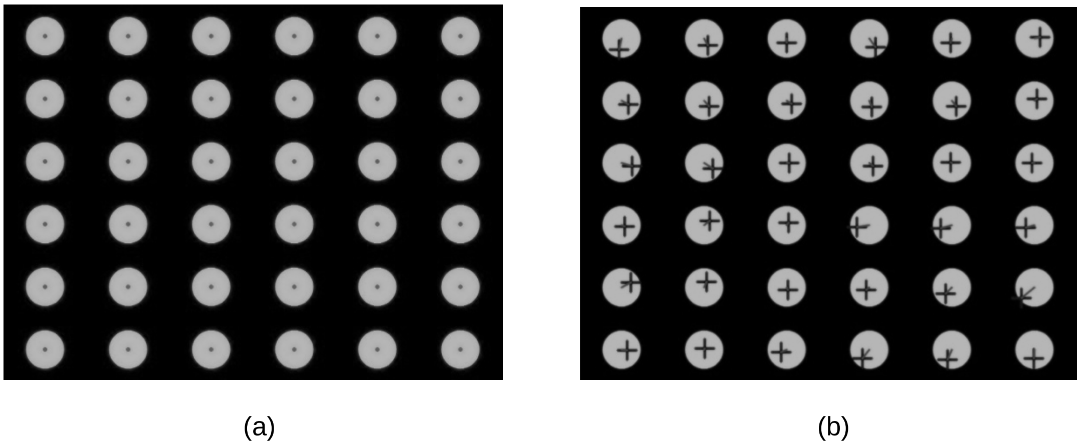


Figure 8. (a) 36 point testing grid, (b) testing result from combination of 25 sample point and third order regression polynomial.

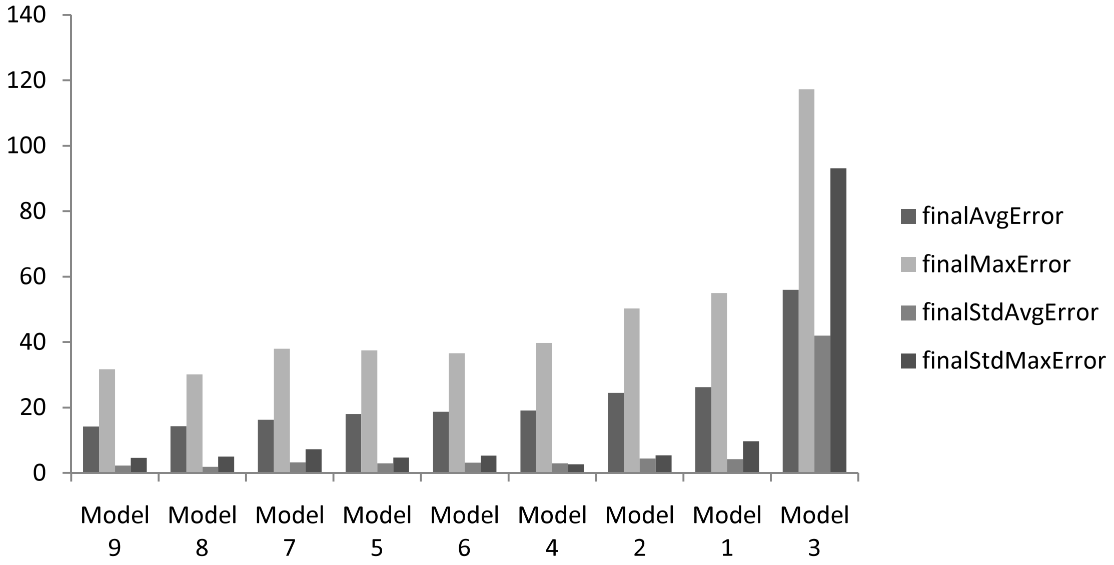


Figure 9. Comparation of 9 testing model

The graphic is shown in Figure 9 has been sorted based on the lowest accuracy from left to right. Sorting process uses average value from 4 indicator which has been told before. The graphic tell us several things below:

1. In general all models except model 3 give a high enough accuracy based on finalAvgError result. This accuracy range from 26.2 until 14.2 pixel or 0.73^o until 0.39^o in visual degree. This result relatively closed to the result in [7], namely about 0.327^o.
2. The best model from left to right are model 9, 8, 7, 5, 6, 4, 3, 2, 1, and the last 3.
3. Model that use more sample points in the calibration process give more accurate result. This can been seen from the top three model are 9, 8 and 7 uses 25 points, then the three model after that are model 5 , 6 , and 4 uses 16 sample points, and the last three model are model 2, 1, and 3 uses 9 sample points.
4. The second and third order polynomial (non linier) regression give more accurate than polynomial liner regression (first order). This can been seen from the top three model are 9, 8 and 7, in model 9 and 8 uses third and second order respectively, whereas in model 7 uses first order. Next three model 5, 6 and 4, in model 5 and 6 uses third and second order respectively, whereas model 4 uses first order. There was exception in next three model 2,1

and 3, in model 3 which uses third order give the worst accuracy, this is because the number of sample points uses in model 3 are 9 points which is that sample points smaller than the number of regression coefficients result uses third order (10 coefficients). Main prerequisite from polynomial regression method is the number of sample points has to be greater than the number of regression coefficients result.

5. Choise of number of order in polynomial regression has direct effect to the number of sample points in calibration process. The higher order is used, so the greater number of sample points is required.

4. Conclusion

We have been developed a low cost device for gaze tracking system. The device is built by utilizing of modified webcam in infrared spectrum. A new technique is proposed in this paper to detect the center pupil coordinate based on connected component labeling. The first, second, and third polynomial regression also tried in the experiments to determine the point of gaze. By combination of center pupil coordinate detection method with third order polynomial regression in determining the gaze point, the experiment results show our system has an acceptable accuracy rate with error pixel range between 26.2 to 14.2 pixel or 0.70 ^{o} to 0.39^o in visual degree. The applied of this system on specific application for general public domain is interesting further research area.

References

[1] A. Duchowski. A breadth-first survey of eye-tracking applications. *Behavior Research Methods Instruments and Computers*. 2002; 34(4): 455–470.

[2] R. Jacob. The use of eye movements in human-computer interaction techniques: what you look at is what you get. *ACM Transactions on Information Systems*. 1991; 9(2): 152–169.

[3] P. Majaranta, K. Raiha. *Twenty years of eye typing: systems and design issues.* In ACM Eye Tracking Research and Applications Symposium. New Orleans, Louisiana, USA. March 2002; 1: 15–22.

[4] A. J. Hornof, A. Cavender, R. Hoselton. *Eyedraw: a system for drawing pictureswith eye movements*. In ACM SIGACCESS Conference on Computers and Accessibility. Atlanta, Georgia. October 2004; 1: 86–93.

[5] L. Sibert, R. Jacob. *Evaluation of eye gaze interaction*. In SIGCHI Conference on Human Factors in Computing Systems. Hague, Netherlands. April 2000; 1: 281–288.

[6] V. Tanriverdi, R. Jacob. *Interacting with eye movements in virtual environments*. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. Hague, Netherlands. April 2000; 1: 265–272.

[7] Yang X, Jiande Sun, Ju Liu, Jinyu Chu, Wei Lie, Yuling Gao. *A Gaze tracking scheme for eye-based intelligent control*. Proceedings of the 8th world Congress on Intelligent Control and Automation. Jinan, China. July 2010; 1: 50-55.

[8] Jose Sigut, Sid-Ahmed Sidha. Iris Center Corneal Reflection Method for Gaze Tracking Using Visible Light. *IEEE Transaction on Biomedical Engineering*. 2011; 58(2): 411-419.

[9] Lee HC., Due Thien Luong, Chul Woo Cho, Eui Chul Lee, Kang Ryoung Park. Gaze Tracking system at a distance for controlling IPTV. *IEEE Transaction on Consumer Electronics*. 2010; 56(4): 2577-2583.

[10] Dan Witzner Hansen, Qiang Ji. In the Eye of the Beholder: A Survey of Models for Eyes and Gaze. *IEEE Transaction on Pattern Analysis and Machine Intelligence*. 2010; 32(3): 478-500.

[11] Lawrence H.Yu, Moshe Eizenman. A New Methodology for Determining Point-of-Gaze in Head-Mounted Eye Tracking Systems. *IEEE Transactions on Biomedical Engineering*. 2004; 51(10): 1765-1773.

[12] Zhor Ramdane Cherif, Amine Nait Ali. An Adaptive Algorithm for Eye-Gaze-Tracking-Device Calibration. *IEEE Transactions On Instrumentation and Measurement*. 2008; 57(4): 716-723.

[13] Craig Hennessey, Borna Noureddin, Peter Lawrence. Fixation Precision in High-Spreed Noncontact Exe-Gaze Tracking. *IEEE Transactions on systems, Man, and cybernetics-Part B:Cybernetics*. 2008; 38(2): 289-298.

[14] Craig Hennessey, Peter Lawrence. Noncontact Binocular Eye-Gaze Tracking for Point-of-Gaze Estimation in Three Dimensions. *IEEE transactions on biomedical engineering*. 2009; 56(3): 790:799.

[15] Ioana Bacivarov, Mircea Ionita, Peter Corcoran. Statistical Models of Appearance for Eye tracking and Eye Blink Detection and Measurement. *IEEE Transaction on Consumer Electronics*. 2008; 54(3): 1312-1320.

[16] L. Young, D. Sheena. Survey of eye movement recording methods. *Behavior Research Methods and Instrumentation*. 1975; 7: 397–429.

[17] J.Babcock, J.Pelz. *Building a light weight eye tracking head gear*. In ACM Eye Tracking Research and Applications Symposium. San Antonio, TX, USA. 2004; 1: 109–114,

[18] Li, Donheng, Babcock Jason, J Parkhurst Derrick. *openEyes: a low-cost head-mounted eye-tracking solution*. ACM Eye Tracking Research and Applications Symposium. San Diego, CA, USA. 2006; 1: 95-100.

[19] Kartika Firdausy, Selamat Riyadi, Tole Sutikto, Muchlas. Aplikasi Webcam untuk Sistem Pemantau ruang berbasis Web. *TELKOMNIKA*. April 2008; 6(1): 39-48.

[20] Goujian Shao, Ming Che, Bingyi Zhang, Kunfang Cen, Wei Gao. *A Novel Simple 2D Model of Eye Gaze Estimation*. Second International Conference on Intelligent Human-Machine Systems and Cybernetics. Nanjing, China. 2010; 1: 300-304.

[21] Qiong Zhang, Zhiliang Wang, Jiannan Chi, Pengyi Zhang, Yi Yang. *Design and Calibration for Gaze Tracking System*. 2nd IEEE international conference on information management and engineering. Chengdu, Sichuan, China. April 2010; 5: 221-225.

[22] Abdul Fadlil. An Automatic Identification System of Human Skin Irritation. *TELKOMNIKA*. December 2010; 8(3): 255-264.