

RESEARCH

Open Access

# A 2D eye gaze estimation system with low-resolution webcam images

Ibrahim Furkan Ince<sup>1</sup> and Jin Woo Kim<sup>2\*</sup>

## Abstract

In this article, a low-cost system for 2D eye gaze estimation with low-resolution webcam images is presented. Two algorithms are proposed for this purpose, one for the eye-ball detection with stable approximate pupil-center and the other one for the eye movements' direction detection. Eyeball is detected using deformable angular integral search by minimum intensity (DAISMI) algorithm. Deformable template-based 2D gaze estimation (DTBGE) algorithm is employed as a noise filter for deciding the stable movement decisions. While DTBGE employs binary images, DAISMI employs gray-scale images. Right and left eye estimates are evaluated separately. DAISMI finds the stable approximate pupil-center location by calculating the mass-center of eyeball border vertices to be employed for initial deformable template alignment. DTBGE starts running with initial alignment and updates the template alignment with resulting eye movements and eyeball size frame by frame. The horizontal and vertical deviation of eye movements through eyeball size is considered as if it is directly proportional with the deviation of cursor movements in a certain screen size and resolution. The core advantage of the system is that it does not employ the real pupil-center as a reference point for gaze estimation which is more reliable against corneal reflection. Visual angle accuracy is used for the evaluation and benchmarking of the system. Effectiveness of the proposed system is presented and experimental results are shown.

**Keywords:** low-cost 2D eye gaze estimation, eyeball detection, stable approximate pupil-center detection, robust eye movement detection, deformable template methods

## 1. Introduction

Gaze information plays an important role in point of regard (PoR) which is a proxy for handling the users' attention or intention as an interaction between human and the computer where it can be used as a form of input instead of keyboard and the mouse [1-3]. Every single eye gaze tracking system has different visual angle accuracies which are important for the size of selection targets such as buttons, icons, pictures, and texts [4]. The systems with lower visual angle accuracy are more robust in terms of selection of targets and visual angle accuracy is generally used for benchmarking of the eye gaze systems [5].

Various systems have been described in the literature. Deng et al. [6] present a local deformable template method for locating the eye and extracting eye features.

A system based on a dual-state model for eye tracking is proposed in [7]. Both of these methods require manual initialization of eye location. Witzner et al. [8] model the iris as an ellipse, but the ellipse is locally fitted to the image through an EM and RANSAC optimization process. A system based on feature detection using one camera is suggested by Smith et al. [9]. Nouredin et al. [10] suggest a two-camera solution where a fixed wide-angled camera with rotating mirror directs the orientation of the narrow-angled camera. A practical gaze point detecting system with dual cameras is also proposed by Park et al. [11]. They employ IR-LED illuminators for wide and narrow view camera to overcome the problem of specula reflection on glasses. They attain the 2.89 cm of RMS error between the computed and the real positions. A non-intrusive eye gaze estimation system is proposed by Yoo et al. [12]. Their system allows ample head movements and yields quite accurate results using cross-ratio under large head motion. Artificial neural network (ANN) is also applied into eye gaze

\* Correspondence: jinwoo@ks.ac.kr

<sup>2</sup>Department of Information Communication Engineering, Kyungsoong University, Busan, South Korea

Full list of author information is available at the end of the article

estimation in the literature [13]. This method employs cropped images of eyes for training the activation functions. It requires calibration and training process which is not applicable in everyday life.

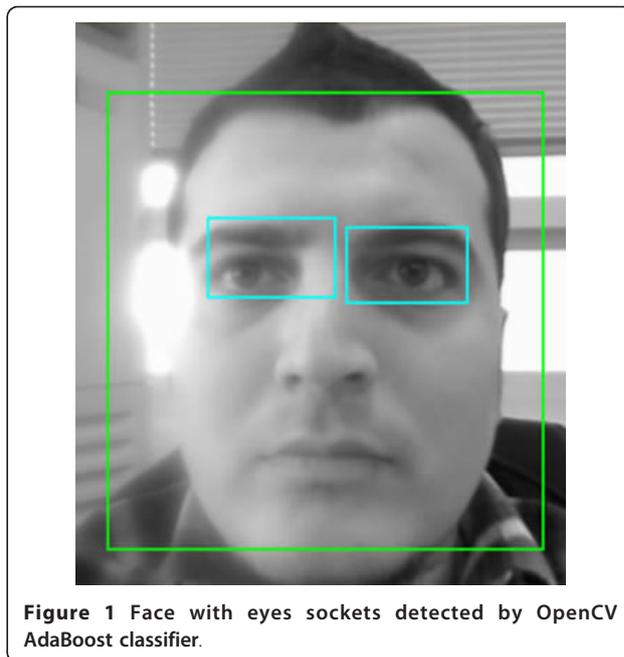
Guestrin et al. [14] claim that the point of gaze (POG) can be estimated only if the head is completely stationary when the system employs one camera and one light source. Therefore, head-movements are the major problem of low-cost 2D eye gaze trackers. This article neglects the head-movements and estimates the approximate POG using shape- and intensity-based methods.

The remaining part of the article is organized as follows. In Section 2, insights of the face and eyes socket detection are presented. In Section 3, deformable angular integral search by minimum intensity (DAISMI) algorithm is introduced. In Section 4, deformable template-based 2D gaze estimation (DTBGE) algorithm with mathematical explanation is described. In Section 5, a dwell-time-based virtual keyboard is employed to measure the visual angle accuracy of the system. Experimental evaluation and benchmarking of the system are given at the end. Section 6 gives some discussion and concludes the paper.

## 2. Face and eyes socket detection

The fundamental requirement of a simultaneous eye tracking and gaze detection system is to accurately detect the eye sockets, which can easily be achieved by Haar-like object detectors which is formerly proposed by Paul Viola [15] and latterly extended by Rainer Lienhart [16]. It allows a classifier trained with sample views of a particular object to be detected in a whole image. The advantages of this method are that when implemented correctly it is fast, efficient, and accurate. It is also effective in detecting objects which are either partially occluded or if the video frame is noisy. By applying it in a multistep detection approach, it is possible to detect precisely the eye region by first recognizing the face and then by setting a region of interest on it. It needs the correct Haar-cascade descriptors for front face and two-eye region. Santana et al. [17] present fully trained Haar-cascade descriptors which are employed in our system and classified by means of Intel OpenCV library. Figure 1 shows a successful screenshot for face and eye socket detection.

The detected eye sockets contain eyeball, eye brow, and eyelashes together. However; in most of the intensity-based trackers, eye brows and eyelashes are described as an obstacle for robust eyeball detection. It is physically known that pupil is the darkest contour in an eye-socket; however, eye brows and eyelashes are also dark and have similar intensity values with pupil. This makes everything difficult and prevents developing robust eye tracking systems.



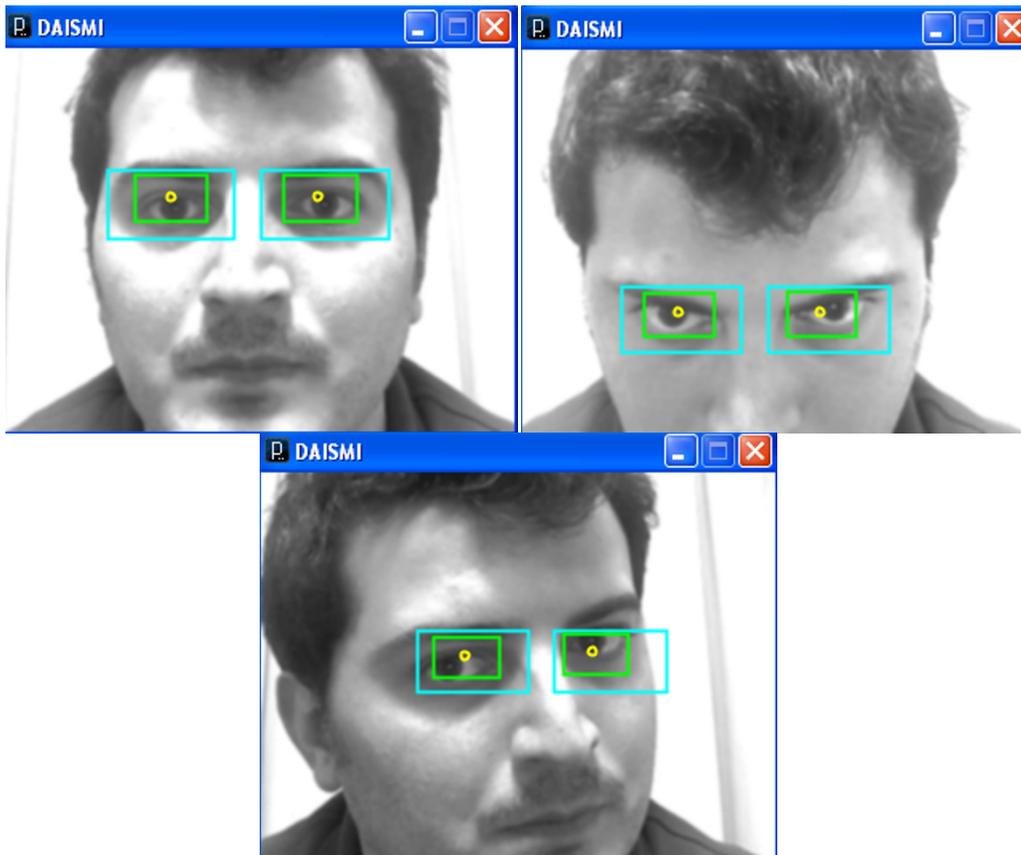
**Figure 1** Face with eyes sockets detected by OpenCV AdaBoost classifier.

The mass-center of an eye-socket is considered as the most probable pupil location in terms of the less weighted intensities [18]. If the mass-center of an eye-socket is consecutively updated, then the resulting center of mass will be much closer to the pupil-center in each loop-step. From this point of view, the mass-center of an eye-socket is derived in  $x$  number of steps in terms of size and position. Gaussian blurring is employed before the processing. Size of the socket is reduced by  $p$  percentage rate in each step. Our system employs  $x$  as 2 and  $p$  as 15%. Finally, eye-socket is reduced in size which mostly excludes the eye-brows. Equation 1 describes the center of mass estimation as follows:

$$C = \frac{\sum_{k=0}^n P_k \times (255 - I_k)}{\sum_{k=0}^n (255 - I_k)} \quad (1)$$

where  $c$  is the position of mass-center,  $P$  is the position, and  $I$  is the intensity of each pixel within the eye-socket which contains  $n$  number of pixels. Figure 2 shows screenshots from the socket reduction process.

The sockets derived from the original ones show more precise location of the eyeball. However, center of mass mostly overlaps on the eyelashes which are still very close to the eyeball. Finally, the derived sockets are employed as eye estimates in which DAISMI is employed to detect the stable approximate pupil-center position by averaging the positions of eyeball border vertices which is necessary for initial template orientation of DTBGE and the final movement displacement vector.

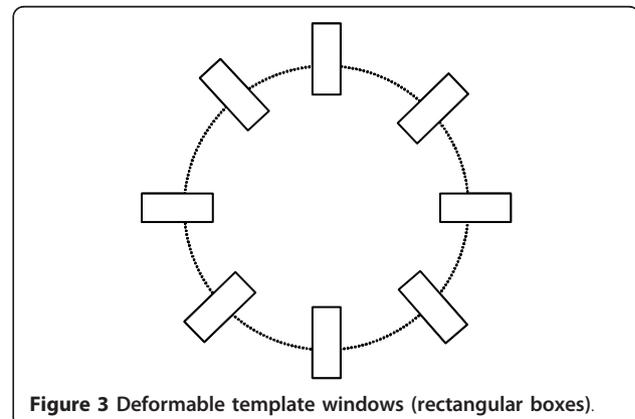


**Figure 2** Socket-size reduction by center of mass approach (blue rectangle shows the original socket and green rectangle shows the derived form after two steps. Yellow circle shows the position of mass-center).

### 3. Deformable angular integral search algorithm by minimum intensity

The shape of eyeball is the most commonly used feature because it is a perfect circle by its nature. Shape-based methods are considered good at handling shape, scale, and rotation changes [19]. Another concern is the intensity features of the eyeball in an eye model. Color feature of an eyeball is known as it is the densest dark contour in an eye estimate. This article proposes DAISMI algorithm which searches the densest dark contour in an eye estimate. Linear-search algorithms are used conventionally in color-trackers. However, they are weak in eyeball detection because of the corneal reflection and varying illumination. Since eyeball is circular, the noise distribution due to varying illumination avoids success when linear-search is applied to the color-tracking system. DAISMI models the shape information of the eyeball and tracks the possible eyeball contours in an angular way by means of deformable template windows. Figure 3 shows the basic configuration of the eyeball template. Eyeball shape is modeled as the eyeball borders are the locations where there is the maximum

amount of variance in the intensity and the intensity distribution in the entire template is homogenous. Model also assumes that border pixels have lower intensity values than the average intensity of the derived eye-socket. First, template is oriented on each derived eye-socket. Literally, the center of the eyeball template should locate over the eyeball contour. Therefore,



**Figure 3** Deformable template windows (rectangular boxes).

darkest pixel in the derived eye-socket is searched by simple linear-search algorithm. The resulting pixel location is considered as initial pupil-center location in derived eye-socket. However, the position of the darkest pixel in derived eye-socket is noisy which means it varies by position due to varying illumination. This is known as the basic problem of all the low-cost eye trackers. DAISMI does not aim to find the real pupil-center, but rather; it finds the mass-center of border vertices by constructing a deformable template which originates from the initial pupil-center position that is estimated by finding the darkest pixel in derived eye-socket with simple linear-search algorithm. Considering the particular conditions of the model, proposed algorithm finds the locations of border vertices which have the maximum integral of average low-intensities through a template window. However, DAISMI does not start searching from the template origin, but rather it starts searching from the last index of the template window to the origin's index. In other words; it searches from outer to inner in each window. After all, the index of the pixel which makes the maximum variation with the previous index through each window in terms of intensity is calculated using a cut-off parameter. Cut-off parameter determines the place of initial possible index which makes the maximum variation with the previous index. This parameter is used for getting rid of eyebrows. In each step of this process, the maximum integral of the average low intensities of the pixels from the resulting index to the origin's index is found. The index number with the maximum integral of average low intensities is considered as a vertex point of the eyeball border. The same process is employed for all other windows in the template. It is expected that model fits to the real eyeball in the derived eye-socket depending on the number of windows. Usage of more windows yields more matching probabilities between the model and real eyeball. Finally, mass-center of resulting border vertices is calculated in terms of low-intensities. The resulting mass-center position is considered as stable approximate pupil-center position and four border vertices which locate at the farthest distance from the mass-center, horizontally, and vertically are considered as the end-points of minor and major axis of the ellipse. Table 1 shows the pseudo-code of the algorithm.

Figure 4 shows successful snapshots of eyeball detection by DAISMI in which number of windows is employed as 80 and window size is employed as the 25% of the derived eye-socket's height while the tolerance value is set to 0.15f. The cut-off ratio is set to 1.25f.

The success rate of fitting depends on four parameters: number of windows, window size, tolerance, and cut-off ratio for the arbitrary model. The larger window

**Table 1 Pseudo code of DAISMI**

---

```

LET S be the rectangular socket region consists of P number of pixels:  $S_1 \dots S_P$ 
LET A be a two-dimensional array of pixels
LET O be the position of the template's origin
LET AVG be the average intensity of socket region
LET W be the deformable eyeball model involves N number of template windows:  $W_1 \dots W_N$ 
LET Z be the size of the template window
LET L be the template window consists of Z number of pixels:  $L_1 \dots L_Z$ 
LET V be the index number of the window where window has the highest intensity variance between current index and previous index.
LET WIDTH be the width of the input camera image
LET alpha be the angle between windows
LET cutOffRatio be the ratio determines the initial index with the highest variance
in a window which is used for getting rid of eyebrows
LET epsilon be the tolerance value for eyeball detection
LET B be the index of vertex over the border of eyeball
SET minIntensity TO 256.0f
FOR EACH i = 1 TO P
    //Start of pixels
    IF BRIGHTNESS(  $S_i$  ) < minIntensity THEN
        SET minIntensity TO BRIGHTNESS(  $S_i$  )
        SET  $O_x$  TO POSITION(  $S_i$  )-> x
        SET  $O_y$  TO POSITION(  $S_i$  )-> y
    END IF
    //End of pixels
END LOOP
SET alpha TO 0.0f
SET integralIndices TO 0.0f
SET integralBrightness TO 0.0f
FOR EACH j = 1 TO N
    //Start of windows
    INCREASE alpha BY (  $TWO\_PI/N$  )
    SET maxVariance TO 0.0f
    SET maxAverageIntensity TO 0.0f
    FOR EACH k = Z TO 1
        //Start of indices in windows
        SET point_x TO  $O_x + k * COS( alpha )$ 
        SET point_y TO  $O_y + k * SIN( alpha )$ 
        SET index TO point_y * WIDTH + point_x
        SET  $A[j,k]$  TO COLOR(  $S_{index}$  )
        SET variance TO ABS( BRIGHTNESS(  $A[j,k]$  ) - BRIGHTNESS(  $A[j,k-1]$  ) )
        SET V TO Integer.MAX_VALUE //highest possible value of an integer.
    END IF
    IF variance > maxVariance AND V > Z / cutOffRatio THEN
        SET maxVariance TO variance
        SET V TO k
    END IF
END LOOP
SET integralIntensity TO 0.0f
    
```

**Table 1 Pseudo code of DAISMI (Continued)**

```

FOR EACH  $i = V$  TO 1
//Integrate from the index with the highest variance to the origin's index
    INCREASE integrallIntensity BY ( 255 - BRIGHTNESS(  $A[i, 1]$  ) )
END LOOP
SET averageIntensity TO ( integrallIntensity / $i$ )
    IF averageIntensity > maxAverageIntensity THEN
        SET maxAverageIntensity TO averageIntensity
        SET  $B$  TO  $V$ 
    END IF
//End of indices in windows
END LOOP
INCREASE integralIndices BY  $B$ ;
INCREASE integralBrightness BY BRIGHTNESS(  $A[i, B]$  )
IF integralIndices/ $j$  > epsilon AND integralBrightness / $j$  < AVG THEN
    ADD  $B$  TO ARRAYLIST
END IF
//End of windows
END LOOP
    
```

size causes larger elliptical fittings with higher accuracy in terms of pupil-center detection while the center of the ellipse is considered as the pupil-center. In addition, it is more probable to attain more fittings for the model when the number of windows is increased. Figure 5 shows successful snapshots of DAISMI after ellipse completion.

#### 4. DTBGE algorithm

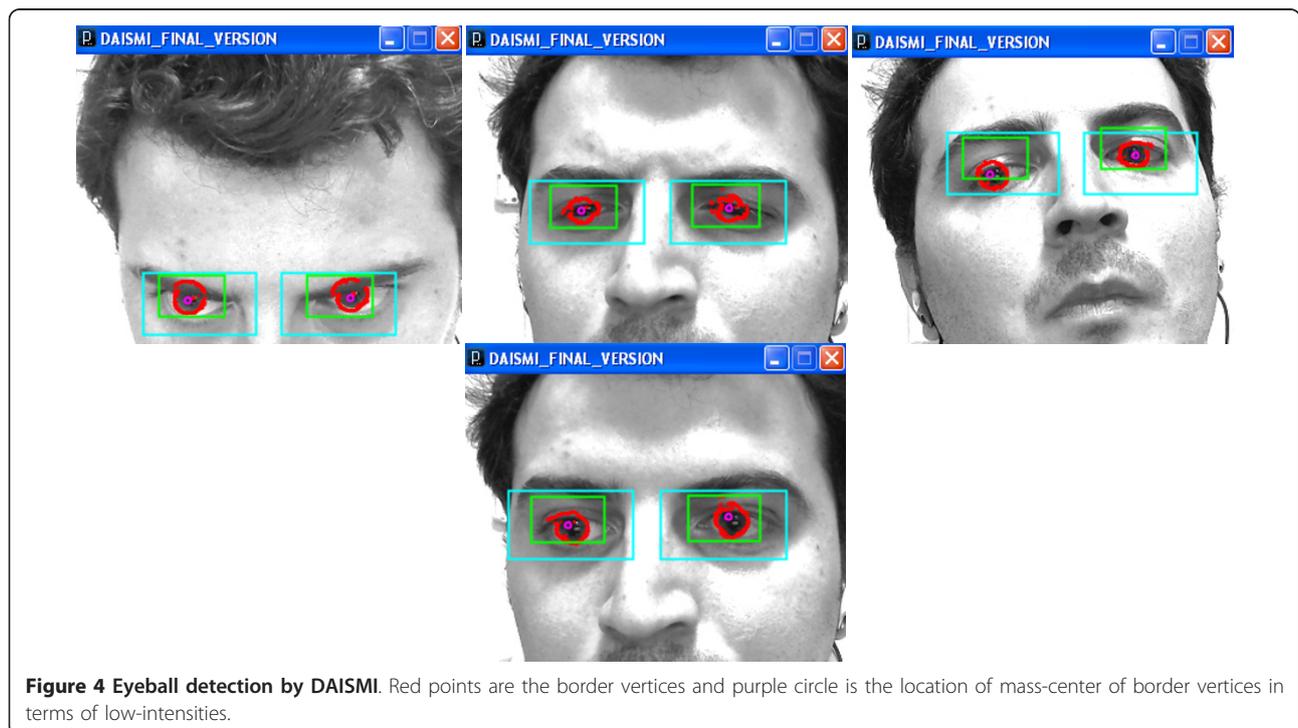
In this section, a binary deformable eyeball template is modeled and 2D gaze estimation is performed depending on the displacement in eye movements. At any horizontal and vertical eye movement, the displacement vector is estimated with respect to the previous frame.

The algorithm consists of four steps as follows:

##### 4.1 The basic configuration

The algorithm aims to decide on the direction of the eyeball movement using a deformable template as shown in Figure 3. In other words; DTBGE is a kind of noise filter which avoids incorrect decisions caused by noisy pupil-center coordinates. DAISMI finds the approximate window size which is basically considered as the Euclidean distance between template origin and eyeball border vertices. DTBGE creates another template for gaze direction detection using the window size estimated by DAISMI and converts the source image into binary using p-tile thresholding algorithm which automates the thresholding ratio frame-by-frame depending on the varying density ratio between black and white pixels over the detected eyeball. Figure 6 shows the orientation of windows on binary source image.

Template is created with eight windows though it may vary depending on the desired accuracy rate. Considering the memory workload and speed of the system, the best results are taken with eight windows though higher number of windows is more precise for decision making.



**Figure 4 Eyeball detection by DAISMI.** Red points are the border vertices and purple circle is the location of mass-center of border vertices in terms of low-intensities.



Figure 5 Ellipse completion by DAISMI.

In this regard; the internal energy of a single window is defined as follows:

$$E_w = \frac{\sum (\text{intensity value of each pixel within the window})}{\text{number of pixels in the window}} - \frac{255}{2} \quad (2)$$

According to Equation 2, black pixels with '0' intensity has a negative impact on the internal energy of a single window while the white pixels with '255' intensity affect the result positively.

#### 4.2 Adjustment of template size

According to the definition of internal energy of a single window in Equation 2, the size of the template varies depending on the angular integral of window energies with angle  $\theta$ .

$$E_{\text{resize}} = \int_0^{2\pi} E_w(\theta) d\theta \quad (3)$$

According to the Equation 3, the size of the template increases when the angular integral of window energies is less than '0' and decreases when it is greater than '0'.

Figure 7 shows a deforming eyeball model in two cases: shrinking and expanding. Decision is made according to Equation 3. In both cases; eyeball size is updated and a new template is created in each frame using the updated window size.

#### 4.3 Horizontal movement of the object

Decision on the horizontal movement of deformable template is made by cosine functions (Figure 8). Cosine functions produce negative values when the angle  $\theta$  is between  $90^\circ$  and  $270^\circ$ . Using Equation 4, the direction of the horizontal movement is determined.

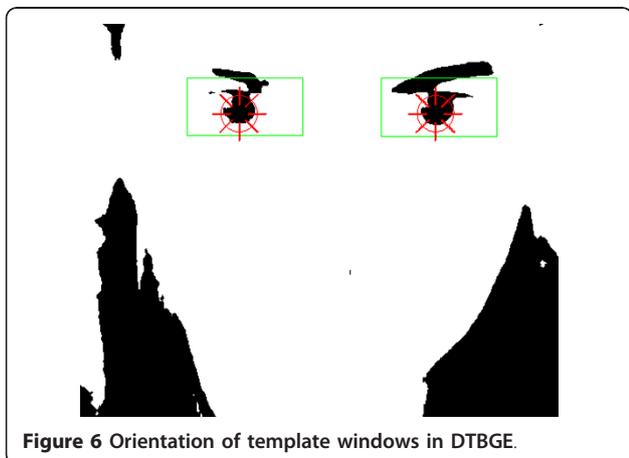


Figure 6 Orientation of template windows in DTBGE.

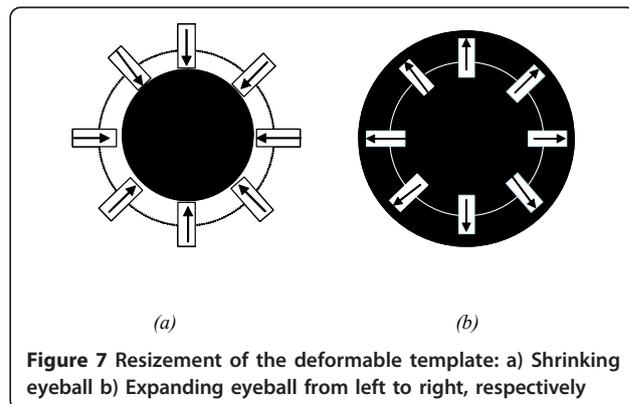
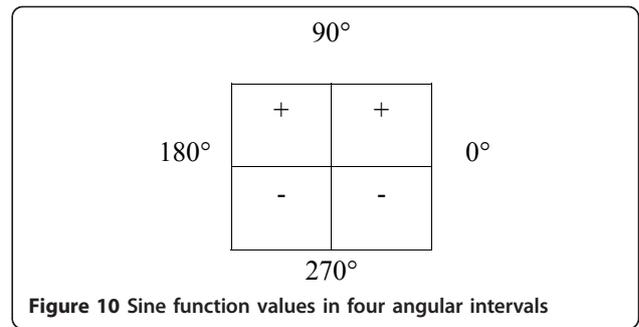
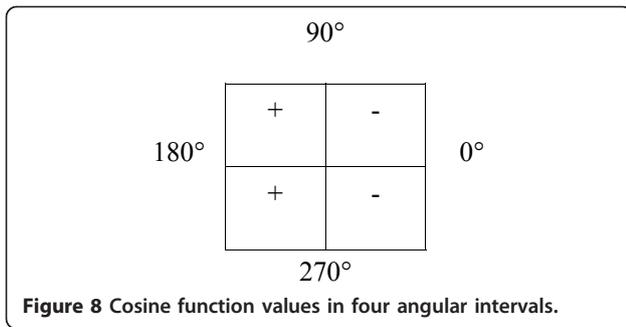


Figure 7 Resizing of the deformable template: a) Shrinking eyeball b) Expanding eyeball from left to right, respectively



$$E_{\text{move}(x)} = \int_0^{2\pi} E_w(\theta) \cos \theta d\theta \quad (4)$$

According to Equation 4, the decision on the horizontal movement is made as to the right-hand side when the angular integral of cosine functions of each window's internal energy is less than '0', conversely (Figure 9); it is to the left-hand side when it is greater than '0'.

$E_{\text{move}(x)} > 0$ , the decision is made to ( $\leftarrow$ ) direction.  
 $E_{\text{move}(x)} < 0$ , the decision is made to ( $\rightarrow$ ) direction.  
 Finally, horizontal movement is pumped.

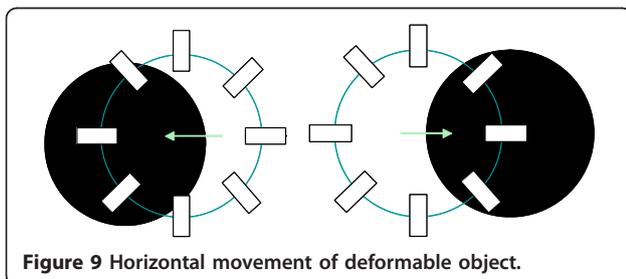
#### 4.4 Vertical movement of the object

Decision on the vertical movement of deformable template is made by sine functions (Figure 10). Sine functions produce positive values when the angle  $\theta$  is between  $0^\circ$  and  $180^\circ$ . Using Equation 5, the direction of the vertical movement is determined.

$$E_{\text{move}(y)} = \int_0^{2\pi} E_w(\theta) \sin \theta d\theta \quad (5)$$

According to Equation 5, the decision on the vertical movement (Figure 11) is made as to up-direction when the angular integral of sine functions of each window's internal energy is less than '0'; conversely, it is to down-direction when it is greater than '0'.

$E_{\text{move}(y)} > 0$ , the decision is made to ( $\downarrow$ ) direction.  
 $E_{\text{move}(y)} < 0$ , the decision is made to ( $\uparrow$ ) direction.



Finally, vertical movement is pumped.

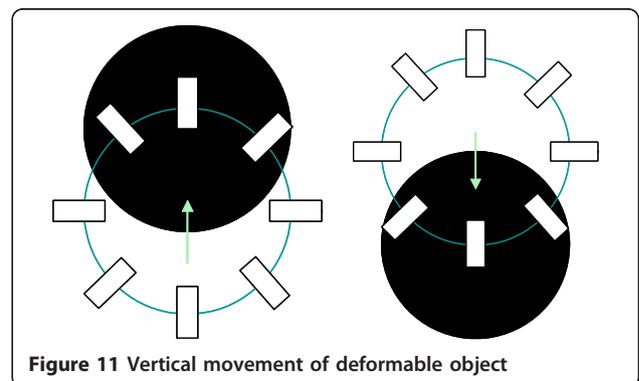
The calibration procedure is performed as a promise between users and system. All the users are asked to look at the center of screen before the tracking starts. After the tracking starts, displacement vector in terms of stable approximate pupil-center position is created by DAISMI. DTBGE decides on the reliable movements in case of noisy displacement vectors. When the users move their eyes, the amount of pixel-wise pupil displacement filtered by means of DTBGE is oriented to the screen relative to the initial gaze point and screen size. The system assumes that the initial position of gaze is the center of screen.

### 5. Experimental results

The system is primarily developed and tested on a Windows XP PC with an Intel Pentium Dual CPU (T2390) with 1.86 GHz processor and 1.75 GB RAM. The monitor size is 12.1" (307 mm) with 9.72" (247 mm) width and 7.28" (185 mm) height with 4:3 aspect ratios. Video is captured with a Logitech Quickcam Pro 4000 webcam at 30 frames per second. Video is processed as binary images of  $160 \times 120$  pixels using various utilities from the Intel OpenCV library.

#### 5.1 Eye tracking experiment

The experiment is conducted for testing the eye tracking system whether it can successfully locate the pupils of the users under different conditions. A dataset



consisting of 300 images taken under varying lighting conditions, head positions, and with complex backgrounds is collected. The experimental results show that the performance rate can reach 100% at frame per second (FPS) = 15 while the camera input image size was  $160 \times 120$ . Figure 12 illustrates some of the successful examples.

### 5.2. 2D gaze estimation and mouse control experiment

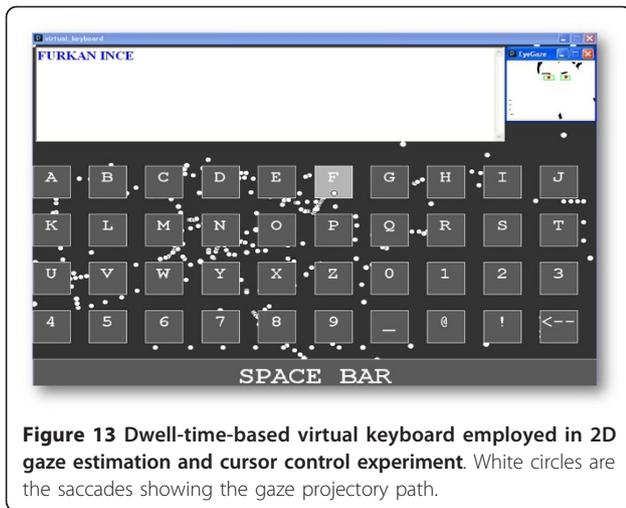
The second experiment is performed by asking six users to move cursor on virtual keyboard buttons (Figure 13) which are generated relative to the screen size. Each user is given a brief introduction about how the system works and starts the experiment without any practice. Each user conducts the experiment five times (Table 2).

When the users are asked to gaze expected buttons, they might look at any pixel over the button. Therefore, it is impossible to measure the gaze accuracy in pixel-wise paradigm. Besides, the accuracy rate depends on the size of the selection targets which are buttons in this case. If the size of buttons is increased, then it is easier to hit the button, and then the accuracy of the system increases. In order to overcome this challenge, the relative unit error distance values which are

calculated with respect to the true hits on buttons are used. If the unit error distance is '0', then it means that user hits the expected button successfully. If it is '1', then it means that user hits the neighbour button of the expected button. Then, the gaze error is assumed as 1 button. And if the unit error distance is ' $\sqrt{2}$ ', then it means that user hits the button which is at the corner of expected button (horizontally: 1 and vertically: 1 button far from the expected button). Lastly, if the unit error distance is ' $\sqrt{3}$ ', then it means that user hits the button which is also at the corner of expected button but in this case it is horizontally: 1 and vertically: 2 (or the reverse) button far from the expected button. According to the experimental results in Table 2 users can hit the expected button in different durations. Durations do not show the elapsed time between initial gaze button and expected gaze button, but rather they show the total elapsed time starting from the tracking system, move the cursor to initial gaze, and move to expected button. In fact, the elapsed time between initial gaze button and expected button is in millisecond which is ignored. Users are observed that they are in difficulty to bring the cursor to initial gaze button. However, even they cannot succeed in the true hit, their resulting gaze



Figure 12 Results of DAISMI and DTBGE algorithms.



**Figure 13** Dwell-time-based virtual keyboard employed in 2D gaze estimation and cursor control experiment. White circles are the saccades showing the gaze projectory path.

is so close to the expected gaze in terms of unit error distance. Therefore, unit error distance values are not employed in terms of gaze accuracy evaluation. That's why most of the gaze trackers are evaluated with respect to their visual angle accuracy. The visual angle is the angle a viewed object subtends at the eye, usually stated in degrees of arc [20]. Figure 14 shows how to measure the visual angle of a subject's eye looking at an object has a certain distance from the eye.

The visual angle  $V$  is measured directly using a theodolite placed at point  $O$  and is calculated using the formula [20]:

$$V = 2 \times \arctan(S/2D) \quad (6)$$

In order to measure the visual angle accuracy for the system, the minimum true saccadic displacement depending on users' eye movements is needed. For our system, according to the experimental results in Table 2, minimum saccadic displacement is 1 unit button.

In our virtual keyboard application, 1 unit button's size is 75 (px) × 75 (px). However, this size is relative to the screen size and the resolution. In order to find the real size in millimetre, the pixel unit is converted to millimetre unit, firstly.

According to the system settings in Table 3 and the formulas in Equations 6-8, visual angle is estimated. It cannot be calculated precisely because it is assumed that users' distance to the monitor is 400 mm and the minimum saccadic displacement is 1 unit button. These inputs are not precise but very close to the reality.

Finally, horizontal and vertical visual angle are estimated. According to Table 4, the horizontal visual angle is 2.07° while the vertical visual angle is 2.48°.

**Table 2** Results of 2D gaze estimation and cursor control experiment

Button ID	EG	RG	UED	Duration(s)
User 1				
1st Button	L	B	1	38.7
2nd Button	2	2	0	23.4
3rd Button	4	V	$\sqrt{2}$	12.9
4th Button	P	P	0	54.2
5th Button	X	Z	1	43.5
User 2				
1st Button	F	F	0	23.3
2nd Button	O	Z	1	24.7
3rd Button	4	4	0	46.1
4th Button	SBAR	SBAR	0	14.3
5th Button	3	←	1	25.8
User 3				
1st Button	R	G	$\sqrt{2}$	49.1
2nd Button	V	A	$\sqrt{5}$	34.7
3rd Button	9	SBAR	1	15.3
4th Button	4	W	$\sqrt{5}$	21.9
5th Button	D	D	0	19.3
User 4				
1st Button	8	7	1	57.6
2nd Button	M	W	1	38.8
3rd Button	5	5	0	42.2
4th Button	T	1	$\sqrt{5}$	19.5
5th Button	E	F	1	26.7
User 5				
1st Button	X	X	0	35.1
2nd Button	F	F	0	24.0
3rd Button	I	R	$\sqrt{2}$	34.6
4th Button	Y	N	1	33.9
5th Button	6	7	1	25.3
User 6				
1st Button	C	M	1	31.1
2nd Button	R	Q	1	28.6
3rd Button	S	H	$\sqrt{2}$	13.8
4th Button	W	V	1	18.7
5th Button	O	P	1	47.7

EG, expected gaze; RG, resulting gaze; UED, unit error distance; SBAR, spacebar.

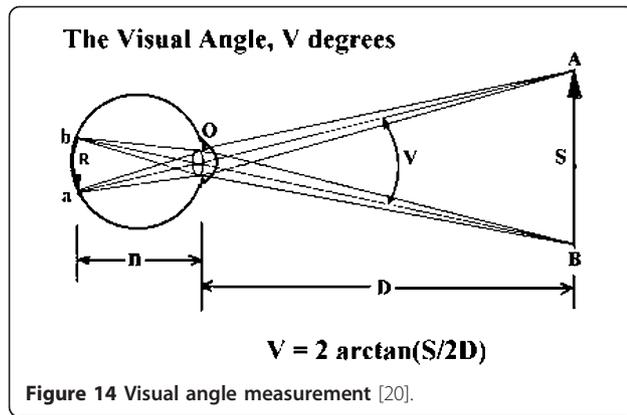
$$\text{Horizontal Visual Angle} = \arctan\left(\frac{(\text{minimum horizontal saccade displacement}/2)}{(\text{distance to the monitor} \times \text{screen resolution width}/\text{screen dimensions width})}\right) \times 2 \quad (7)$$

$$\text{Vertical Visual Angle} = \arctan\left(\frac{(\text{minimum vertical saccade displacement}/2)}{(\text{distance to the monitor} \times \text{screen resolution height}/\text{screen dimensions height})}\right) \times 2 \quad (8)$$

Finally, the visual angle accuracy of the system is considered as around 2°.

## 6. Conclusions and discussions

In this article, a shape- and intensity-based deformable eye pupil-center detection and movement decision algorithms



are introduced in terms of developing a low-cost 2D gaze estimation and cursor control system regardless of the real position of pupil-center. In many other algorithms which are based on PoR information, real pupil-center position is required to get rid of noisy eye movements. However, unlikely, DTBGE employs a new deformable template model which decides on the eye movements' directions in binary source images regardless of knowing the real pupil-center. DAISMI finds the stable approximate pupil-center location by searching deformable eyeball models which includes low-intensity pixels within the template windows. It finds that the position of the eyeball with border vertices and the mass-center of vertices are considered as the stable approximate pupil-center position. Even the origin of the template is vibrating and noisy, the border vertices are almost stable and less vibrating. The stable approximate position of pupil is enough for getting robust eye movement detection and corresponding pupil-center displacement. This system is very useful for those who want to build 2D gaze estimation and mouse control systems with low-resolution source images such as simple webcams without infra-red filter. Experimental results show very encouraging results such that the visual angle accuracy is around 2° while the speed is 15 fps.

The comparison of the system with other systems in the same category is as follows:

As seen in Table 5, our system has no calibration, no training process, no infra-red filtering with head-mounted camera, and it is the only system which does

**Table 3** System settings for the experiment

Screen resolution	
Width	1280 pixels
Height	800 pixels
Screen dimensions (1 inch = 2.54 cm)	
Width	247 mm
Height	185 mm
Subject	
Distance to the monitor	400 mm

**Table 4** Calculation of visual angle accuracy

What is the angle for minimum saccade displacement?		
Horizontal	Saccade width	75 pixels
	Angle	2.072829°
Vertical	Saccade height	75 pixels
	Angle	2.48392°

**Table 5** Comparison of the system with several main-stream 2D gaze estimation systems

Infra red	Training and calibration	Real pupil-center based	Camera and video size	Accuracy (deg)	Algorithms
Yes	No-No	Yes	Head-mounted (320 × 240)	2-4	Active contours [8]
Yes	No-Yes	Yes	Head-mounted (320 × 240)	2.9	Image processing [10]
No	Yes-No	Yes	Head-mounted (320 × 240)	1.5	ANN [13]
No	No-No	No	Webcam (160 × 120)	2	DAISMI DTBGE

not need real pupil-center position for 2D gaze estimation. Even the system employs low-resolution webcam images (160 × 120), the success rates are so close to each other. For this reason, this system is an ideal option for those who cannot afford expensive trackers, does not like calibration, and wants to get high accuracy even the system is with a simple webcam. Finally, this system can be used with more performance if the auto-thresholding can be provided more robustly. The ratio between white and black pixels in template windows is the only decider parameter which determines the performance of the system. However, this ratio depends on the thresholding algorithm only. Better thresholding algorithms can yield better results.

**Abbreviations**

ANN: artificial neural network; DAISMI: deformable angular integral search by minimum intensity; DTBGE: deformable template-based 2D gaze estimation; FPS: frame per second; POG: point of gaze; PoR: point of regard.

**Author details**

<sup>1</sup>School of Engineering, The University of Tokyo, Tokyo, Japan <sup>2</sup>Department of Information Communication Engineering, Kyungsung University, Busan, South Korea

**Competing interests**

The authors declare that they have no competing interests.

Received: 18 January 2011 Accepted: 12 August 2011  
Published: 12 August 2011

## References

1. RJK Jacob, KS Karn, *Eye Tracking in Human-Computer Interaction and Usability Research: Ready to Deliver the Promises, in the Mind's Eye: Cognitive and Applied Aspects of Eye Movement Research* (Elsevier Science, Amsterdam, 2003), pp. 573–605
2. S Zhai, What's in the eyes for attentive input. *Commun. ACM.* **46**(3), 34–39 (2003). doi:10.1145/636772.636795
3. M Kumar, J Klingner, R Puranik, T Winograd, A Paepcke, Improving the accuracy of gaze input for interaction, in *ETRA'08, ACM, New York, USA*, pp. 65–68 (2008)
4. CH Morimoto, RM Mimica, Eye gaze tracking techniques for interactive applications. *Comput Vis Image Understand.* **98**(1), 4–24 (2005). doi:10.1016/j.cviu.2004.07.010
5. A Bulling, H Gellersen, Toward mobile eye-based human-computer interaction. *IEEE Pervasive Comput.* **9**(4), 8–12 (2010)
6. J Deng, F Lai, Region based template deformation and masking for eye-feature extraction and description. *Pattern Recogn.* **30**(3), 403–419 (1997). doi:10.1016/S0031-3203(96)00086-6
7. Y Tian, T Kanade, J Cohn, Dual-state parametric eye tracking, in *Proc of Conf on Automatic Face and Gesture Recognition*, 110–115 (2000)
8. Witzner D Hansen, AEC Pece, Eye tracking in the wild. *Comput. Vis Image Understand.* **98**(1), 182–210 (2005). doi:10.1016/j.cviu.2004.07.014
9. P Smith, M Shah, N Da Vitoria Lobo, Determining driver visual attention with one camera. *Trans Intell Transport Syst.* **4**(4), 205–218 (2003). doi:10.1109/TITS.2003.821342
10. B Nouredin, PD Lawrence, CF Man, A non-contact device for tracking gaze in a human computer interface. *Comput Vis Image Understand.* **98**(1), 52–82 (2005). doi:10.1016/j.cviu.2004.07.005
11. KR Park, J Chang, MC Whang, JS Lim, DW Rhee, HK Park, Y Cho, Practical gaze point detecting system, in *DAGM-Symposium*, 512–519 (2004)
12. DH Yoo, MJ Chung, A novel non-intrusive eye gaze estimation using cross-ratio under large head motion. *Comput Vis Image Understand.* **98**(1), 25–51 (2005). doi:10.1016/j.cviu.2004.07.011
13. S Baluja, D Pomerleau, Non-intrusive gaze tracking using artificial neural networks. *Adv Neural Inf Process Syst.* **6**, 753–760 (1994)
14. ED Guestrin, M Eizenman, General theory of remote gaze estimation using the pupil center and corneal reflections. *IEEE Trans Biomed Eng.* **53**, 1124–1134 (2006). doi:10.1109/TBME.2005.863952
15. P Viola, M Jones, Robust real-time face detection, in *Proc Int Conf on Computer Vision.* **II**, 747 (2001)
16. R Lienhart, J Maydt, An extended set of Haar-like features for rapid object detection, in *Proc of the IEEE Conference on Image Processing (ICIP'02)*. **1**, 1900–1903 (2002)
17. MC Santana, O Déniz-Suárez, L Antón-Canalís, J Lorenzo-Navarro, Face and facial feature detection evaluation-performance evaluation of public domain HAAR detectors for face and facial feature detection, in *Proceedings of the Third International Conference on Computer Vision Theory and Applications (VISAPP 2008)*, vol. 2. (Madeira, Portugal, January 22-25, 2008), pp. 167–172
18. X Long, A Kiderman, OK Tonguz, A High speed eye tracking system with robust pupil center estimation algorithm, in *Proceedings of the 29th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBS 2007)*, Lyon, France (August 2007)
19. DO Gorodnichy, On importance of nose for face tracking. in *Proceedings of the IEEE International Conference on Automatic Face and Gesture Recognition (FG 2002)*, Washington, DC, pp. 188–196 (May 20-21 2002)
20. Visual angle [http://en.wikipedia.org/wiki/Visual\\_angle](http://en.wikipedia.org/wiki/Visual_angle). Accessed 09 May 2011

doi:10.1186/1687-6180-2011-40

**Cite this article as:** Ince and Kim: A 2D eye gaze estimation system with low-resolution webcam images. *EURASIP Journal on Advances in Signal Processing* 2011 **2011**:40.

**Submit your manuscript to a SpringerOpen® journal and benefit from:**

- Convenient online submission
- Rigorous peer review
- Immediate publication on acceptance
- Open access: articles freely available online
- High visibility within the field
- Retaining the copyright to your article

---

Submit your next manuscript at ► [springeropen.com](http://springeropen.com)

---