Proceedings of the 2009 IEEE International Conference on Systems, Man, and Cybernetics
San Antonio, TX, USA - October 2009

Gaze Tracking: A Sclera Recognition Approach

J. R. Parker
Faculty of Fine Arts
University of Calgary
Calgary, Canada
jparker@ucalgary.ca

A. Q. Duong
Department of Computer Science
University of Calgary
Calgary, Canada
tinoduong@gmail.com

***Abstract* - Gaze tracking is the complex process of detecting where a person is looking. In this paper, we present a gaze tracking system that estimates the eye gaze by using a stable reference point created through the extraction of the sclera – otherwise known as the "whites" of the eye.**

***Keywords* - eye tracking, gaze tracking, vision, image processing.**

I. Introduction

The evolution of computing technology has advanced rapidly since its conception. Over time, computers have steadily grown cheaper, more powerful and more ubiquitous. Unfortunately, one aspect of computing that has experienced limited growth is in the area dealing with how people interact with computers. In an attempt to alleviate the limitations inherent to traditional computer input devices, such as the keyboard and mouse, researchers have looked towards alternate methods.One such method is known as gaze tracking. Gaze tracking is the process of determining where a person is looking.

The problem with current gaze tracking systems is that they all require expensive hardware or artificial environments such as head mounted cameras and chin rests [6]. Therefore, their high cost and intrusiveness make them less attractive to most researchers and practitioners. The ideal gaze tracking system is one that provides high accuracy, robustness, ease of use and cost effectiveness.

This paper introduces new techniques devised to create a gaze tracking system that relies solely on image processing and pattern analysis. Special attention was made to ensure that the cost of the system remained low; therefore, the only piece of extra hardware required is a simple CCD camera. By examining an image taken of a person's face for their sclera – a.k.a, the whites of the eye – a unique pattern is created whereby a stable reference point can be extracted. This reference point is used to calculate the eye gaze of a person sitting in front of a computer monitor.

II. Previous Work

Currently, most gaze trackers are video based systems that operate using information found within an image of the face. To calculate the gaze coordinate, a number of different approaches have been developed. These include artificial neural networks [3,4], point-reference techniques [2,7,10], and 3 dimensional approaches [8]. For these techniques, an accurate extraction of information – or features – used for the calculations is paramount.

To simplify the extraction process many researchers use special devices such as head mounted cameras, chin rests, or infrared light emitting diodes (IR LEDs). The most popular systems to date rely on IR LEDs in order to extract the pertinent features needed for gaze calculation [7,9,10].

In addition to the high cost of specialized hardware, IR LEDs can raise a safety concern surrounding prolonged exposure [5]. This can lead to unwillingness to use an infrared based system. The gaze tracker discussed in this paper was implemented in an attempt to alleviate the drawbacks and limitations previously mentioned. The system is able to avoid the negative effects such as extra costs, complexity and potential safety concerns, while providing accurate results with a reasonable level of robustness.

III. Sclera Detection

There are two features used to calculate the gaze coordinate in our system; namely, the iris center and a stable reference point, called the eye-region point, defined by searching for the sclera region within a face.

The motivating factor behind using the sclera is that the color of a healthy sclera is relatively stable across different genders and ethnic characters. Searching an image for sclera clusters allows the system to locate the eye regions. This facilitates the extraction of the iris center and the eye-region point.

Locating the Eye Region

Borrowing concepts heavily utilized in skin detection, the sclera is extracted by analyzing the image for sclera colored pixels. The sclera regions are extracted in a two phase process. First, the normalized histogram of the face bounding box is examined as a rough filter to eliminate non sclera pixels. Second, the pixels which pass the first phase are subjected to a Mahalanobis distance to be classified as sclera or non-sclera pixels.

Using a bounding box of the face, each pixel is assumed to belong to one of either four subsets: A = {Skin}, B = {Sclera}, C = {Hair, Iris, Eye Shadow} or D = {Noise}. It has been noted by [11] that the skin and sclera have high values in the red color component of an image. Using this knowledge, the sclera segmentation algorithm generates a histogram of the R component in the RGB space to find the skin and sclera.

Pixels in set D are those that belong to noise such as reflection or glare off the nose or forehead. This set is easily removed by searching for pixels where the RGB values appear flushed (all three components are near 255). Pixels in set C are eliminated via an iterative thresholding process. A pixel i is classified if its R component falls below a threshold T_r. In order to accommodate for varying lighting conditions, the threshold T_r is adjusted according to the average intensity of the face bounding box for each frame. If the average intensity is too low, then the threshold is lowered in order to not classify skin and sclera as belonging to set C. The value of T_r is initially seeded at 100, which was found to be a good starting point through experimental analysis.

When set C and D are removed from the face bounding box, the sclera segmenter generates a color histogram for the R component of the nRG color space. This gives us a distinctive pattern that was found to occur across all people. Figure 1 illustrates the distinctive bi-modal histogram, where the larger peak represents the more numerous skin pixels and the smaller peak represents the sclera pixels. Although this pattern is more exaggerated in some than others, it was found that the common trend exists throughout all people. By analyzing the pattern of the R histogram, a set of thresholds that bounds the sclera peak is found and is used to filter out pixels that are far away from the sclera peak.

The sclera detector then enters a second phase, where it uses the Mahalanobis distance to classify pixels that passed the initial thresholding.

For each pixel that passed the initial thresholding stage, the Mahalanobis distance is applied on the (U,V) components in the YUV color space. The (U,V) covariance matrix and mean (U,V) matrix needed for the Mahalanobis distance are created in a training phase and loaded into the system on initialization. The training data is composed of over 2000 sclera samples and over 2000 skin samples, taken from multiple subjects, of various ethnicities, and in various lighting conditions. Figure 2 displays a graph of the sclera and

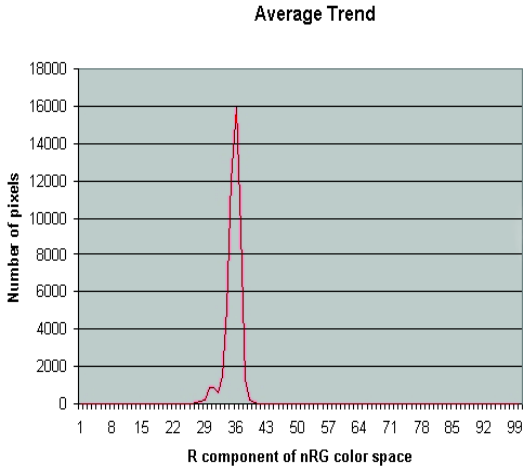


Figure 1: Sclera vs. Skin pixels in nRG

skin training data used for the Mahalanobis distances. Notice how the two sets of pixels group into two distinctive clusters.

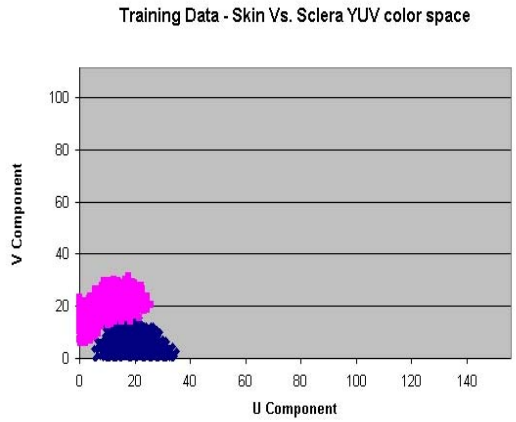


Figure 2: Sclera vs. Skin in YUV

To accommodate changing lighting conditions and slight variances of sclera color from person to person, the threshold for the Mahalanobis distance is dynamically set based on an estimated size of the person's eye. This increases the robustness of the system and reduces the need for manual settings and an extended calibration stage. An example result of the sclera extraction algorithm is given below in Figure 3, where the selected sclera pixels are highlighted grey. The false positives are later removed using basic image processing techniques, in particular using the size of the region and the nature of its neighborhood.

IV. The Eye-Region Point

After the sclera pixels have been located, the iris pixels are found through thresholding and a least squares approximation algorithm. The combination of the sclera pixels and iris circle are used to define the shape of the eye region and a bounding box surrounding it. This eye region is used to create our novel feature called the eye-region point.

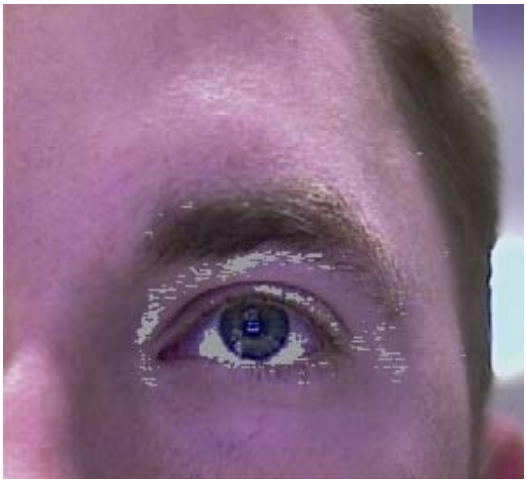


Figure 3: Potential Sclera Pixels

The eye-region point is useful because it is built upon features that have already been extracted; therefore, the extra processing needed is minimized. Not only can the feature be found accurately and consistently, it can also be found quickly. Another major benefit to using the eye-region point is that it remains in a constant position relative to the user's head movement.

There are three main steps to finding the eye-region point

1) Define eye region boundary
2) Define eye region major axis
3) Compute eye-region point

Eye Region Boundary

To find the eye region boundary, the contents of the bounding box are assumed to fit into two different sets: *skin = {Skin and Shadow}* and *eyeRegion = {Sclera and Iris}*. For each pixel i, the Mahalanobis distance is calculated from both the skin and the eyeRegion, where .

$$M_i(i) = \sqrt{(i-\mu)^{\tau}\Sigma^{-1}(i-\mu)}$$

The covariance and mean matrix Σ, and μ, are defined previously. The pixel is classified as $i \in skin$ or $i \in eyeRegion$ depending on which set it is closer to.

Because the iris pixels are darker than the surrounding sclera, we get intermediate sets where the skin, shadow and iris pixels are grouped together, and the sclera is alone. To separate the iris from the skin and shadow, all pixels within the boundary of the iris circle are reclassified as eye region pixels. This yields the desired two sets: skin and eyeRegion. As seen in Figure 4, the eye region, indicated by white, has been separated from the surrounding skin region, shown in black.

To eliminate the false positives that may appear surrounding the true eye region, lines are drawn starting from the center location of the iris outward until a skin pixel is found. Lines will be drawn for every degree going full circle for 360 degrees. The area covered by the lines is now taken as our eye region. In Figure 5, the area highlighted in grey depicts the extracted eye region.

Figure 4: Eye region pixels in white

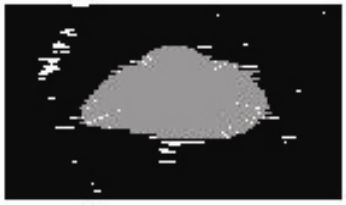


Figure 5: Eye Region Boundary

Although the precise shape of a human eye can vary from person to person, the eye region can be generalized as an ellipse. Finding the line that represents the axis running down the length of the eye is similar to the major axis of an ellipse. Figure 6 illustrates this concept.

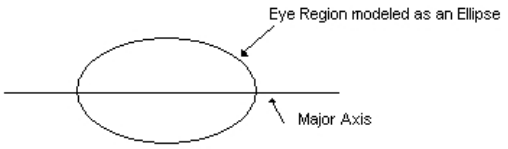


Figure 6: Major axis of eye region

The major axis of the eye region is found by using a least squares line approximation on the shadow pixels that surround the contours of the eye. The least squares approximation assumes the best fit line is one that yields a minimal sum of deviations squared from the line $y=mx+b$ to our N data points (x_i, y_i) , as seen in Equation 1. Therefore, we can use it on our set of shadow pixels to produce a best line that runs through the eye region.

$$\sum_{i=1}^{N} (y_i - (mx_i + b))^2 = Minimal \qquad (1)$$

To obtain the set of eye shadow pixels, basic image processing techniques of equalization and thresholding are applied to a grey scale image within the eye region bounding box. To ensure that the iris pixels are not included within the shadow pixels, all pixels within the iris circle are removed. Figure 7 displays the shadow pixels after the iris pixels have been removed and the line that is created using a least squares line approximation.

Compute Eye-Region Point

Using the eye region boundary and the major axis, the eye-region point is found by calculating the center of mass of the eye region. However, only a section of the eye is used because care must be taken in certain cases where the iris is partially occluded by the top eye lid.

Figure 7: Shadow Pixels with Line

Figure 8: (Left) Contour of Eye Region and Iris while looking up. (right) Eye Region with bulbous effect

Observe in Figure 8 a) that the iris is partially occluded. Note that the iris circle has still been estimated correctly and there is an outline for the occluded section. Due to the nature of the algorithm that estimates the eye region boundary, all pixels within the iris circle are classified as *eyeRegion* pixels. When the iris is partially occluded by the top eye lid, the upper section of the eye region is slightly incorrect because the upper eye lid (i.e. skin) is taken to be part of the eye region. This causes a bulbous effect to occur at the top section of the eye region, as seen in Figure 8 (right).

To remedy this, only the bottom section of the eye region is ever used. As seen in Figure 9, two lines are created using the slope from the major axis. The first line represents the bottom of the eye region and is placed along the bottom contour. The second line represents the upper boundary of the eye region and is used to ensure the bulbous hump is excluded from the center of mass calculations.

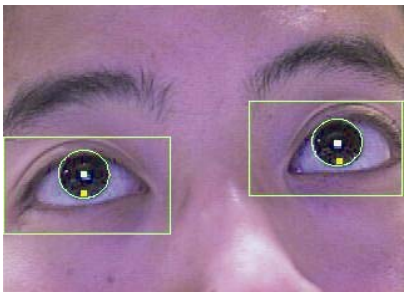
Figure 9: Line boundary

The second line is placed a certain distance from the first line, where the distance is estimated as 20% of the bounding box height. This number was found to be useful through experimental analysis because it was found to always exclude the bulbous hump. Therefore, the eye-region point is calculated by using only pixels of the eye region that have row coordinates that fall in between line one and line two.

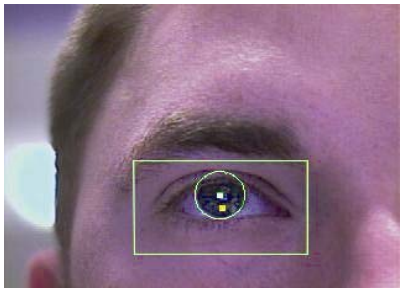
Figures 10 a) and 10 b) show two examples of both the iris center and the eye-region point extracted, where the iris center is in the middle of the iris circle, and the eye-region point is the point below the iris center.

Calculating the Gaze

Our system uses a reference point technique, where the movement of the eye is calculated by comparing the iris center to a stable reference point. In our case, the iris center is used to measure eye movement, and our novel eye-region point is used as the stable reference point. Both features are combined together to create a vector called the *eye-Iris* vector and is defined in Equation 2.



(a)



(b)

Figure 10: Example Output Images

$$V = \{(e_{row}, e_{col}), (i_{row}, i_{col})\} \quad (2)$$

The points (e_{row}, e_{col}) and (i_{row}, i_{col}) are the row and column coordinates of the eye-region point and the iris center respectively. This vector is similar to vectors defined in the work of others who use the reference point as their method of tracking gaze [7,12].

By having the user under go a short calibration stage prior to each system use, template eye-Iris vectors are created as the user fixates on each far corner of the monitor. From these templates, we are able to extract the maximum vertical and horizontal pixel range as the user looks around the screen. Using these ranges a linear interpolation is then performed with subsequent eye-Iris vectors to calculate a gaze coordinate.

For each frame T_i after the calibration, the current eye-Iris vector is extracted and the column displacement is compared against the maximum range derived from the templates. Equation 3 is used to calculate the column coordinate of the eye gaze.

$$Column = \left(\frac{normColDisp}{colRange}\right) \bullet screenWidth \quad (3)$$

The *colRange* is the maximum range of column movement extracted from the templates during calibration, *screenWidth* is the width of the screen in pixels, and *normColDisp* is the amount of column movement normalized for our current eye-Iris vector from.

Similarly, the row coordinate is calculated using Equation 4.

$$Row = \left(\frac{normRowDisp}{rowRange}\right) \bullet screenHeight \quad (4)$$

The *rowRange* is the maximum vertical range, *screenHeight* is the screen height in pixels, and *normRowDisp* is the normalized row displacement between the iris center and the eye-region point for frame.

V. System

Our system is designed for simple use and robustness, so as to allow the user to work as freely as possible. To achieve this goal, as well as avoid complexity and cost, only a single CCD camera is used. The camera is positioned in the front center of the monitor and captures images of a person's face. These images are processed to extract features from the face that are used in the gaze calculation phase. A diagram of the system set up is provided below in Figure 11.

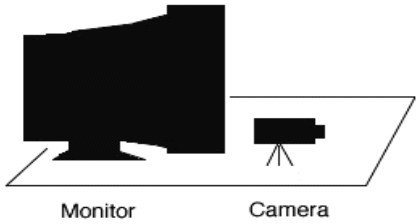


Figure 11: System Setup

Several experiments were performed to test the validity of our technique in both using the sclera detection as a means to locate the eye region and as well as using our novel eye-region point as a reference point for gaze tracking.

Sclera Detection

In our system, sclera detection lies at the root of successful gaze tracking because it is used to find both the iris center and the eye-region point. Within literature, very little research has been done on sclera detection. A single paper called "Active Detection of Eye Scleras in Real Time" was written in 2000 where Betke claimed to be the only one that looked into the problem of sclera detection [1]. To our knowledge, no other significant research has been performed since. To illustrate that our sclera detection technique shows an improvement over the previous work done, our experiments were designed to closely mimic those of Betke's published results.

In the experiments outlined by Betke et al, eye regions are said to be successfully extracted if at least one eye region is located within the image [1]. The standards for our system are higher in that if both eye regions are captured in the image, the sclera detection is only considered successful if both eye regions are located and fully bound. Note: the experiments were performed across people of various ethnicities, genders and lighting conditions.

The first experiment gauges the long term use of the system. Images from the sclera detection were recorded at a constant sampling rate over a period of 15 minutes, where a minimum of 190 images were captured. The results of this experiment are shown in Table 1.

Table 1: Experiment 1 Results

	Betke	**Ours**
Accuracy	63%	95%
Duration	11.5 min	15 min

Table 1: Experiment 1 Results

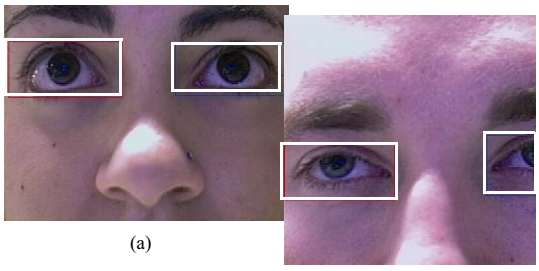
	Betke	Ours
# Frames	190	190
#Subjects	1	3
Success definition	At least 1 eye	All eyes in image

In the second test, the system's performance was evaluated for short term intervals across a broader range of different people. The users sat in front of the camera for 30 second intervals, and images were recorded at a constant sampling rate. The results are seen in Table 2. For this experiment the improvement in accuracy rates are not as significant; however, it should be kept in mind that the definition of success is more rigid in our system.

Table 2: Experiment 2 Results

	Betke	Ours
Accuracy	89	93
Duration	33 sec x 18	33 sec x 18
# subjects	1	3
Success definition	At least 1 eye	All eyes in image

Our results show that the process of eye region extraction via sclera detection has been improved upon. Two output images are given below in Figure 12 a) and b).



(a) (b)

Figure 12: Sample Output

Gaze Tracking Tests

Several tests were performed on different users to evaluate how well the novel eye-region point feature works for gaze tracking purposes. For these tests a closer crop on the eye region was taken in order to increase the resolution of the eye. The test consisted of 3 different people across different ethnicities and lighting conditions.

The first set of tests was performed to evaluate the horizontal and vertical accuracy separately. After the calibration phase, subjects were asked to follow a 25x25 pixel block cursor across the screen. The cursor was designed to move across the screen in a horizontal and vertical motion that covered the outer extremities of the screen – i.e. top to bottom and left to right. The system stored the estimated gaze coordinate along the correlating ground truths. Each test lasted 2-3 minutes and information was stored for each frame. Figures 13 a) and 13 b) show the results of the experiments in a graph. The solid lines are the ground truths and the jagged lines are the gaze coordinate estimates.

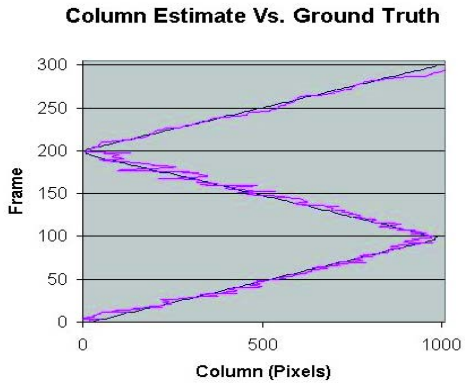


Figure 13: a): Column Ground Truth vs. Column Estimate

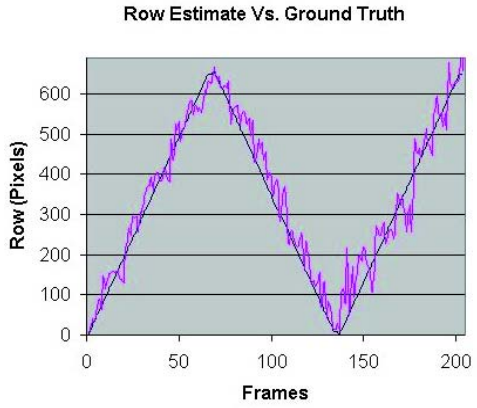


Figure 13: b): Row Ground Truths vs. Row Estimate

An additional test was performed where a cursor was drawn on the screen at 11 random positions and the subjects were instructed to fixate their gaze on each block. The system calculated an estimate of the user's eye gaze and recorded each gaze point along with the corresponding ground truths. Each block remained in a fixed position for a count of 15 frames. Example output of the test is given below in Figure 14, where the large dark squares are the ground truths and the smaller shapes beside them are the estimated gaze coordinates.

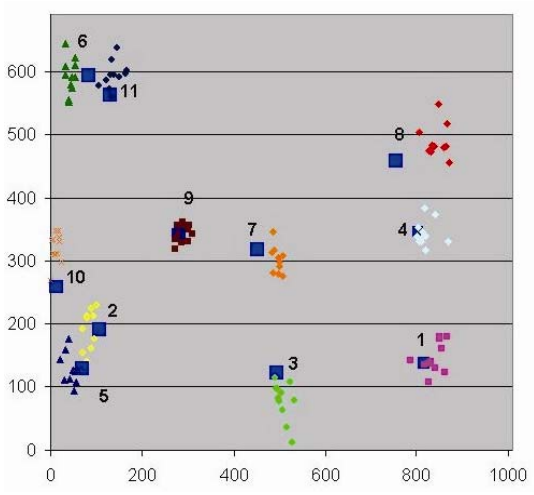


Figure 14: Scaled Representation of a Screen

VI. SUMMARY

Our system was able to calculate a gaze coordinate with an accuracy of 1.5 centimeter radius The results displayed by our system using the novel eye-region point are comparable to ones published in other studies and commercial systems. The system is able to compute the user's gaze coordinate without any prior knowledge of the user's face and without any special hardware other than a CCD camera. The success outlined in our experiments show that the novel eye-region point can be successfully used for gaze tracking purposes.

REFERENCES

[1] Betke, M. Mullally, B., and Magee, J. Active Detection of Eye Scleras in Real Time, *IEEE CVPR Workshop on Human Modeling, Analysis and Synthesis*, Hilton Head Island, South Carolina, June 2000.

[2] Betke, M., and Kawai, J. Gaze Detection via Self-Organizing Gray-Scale Units, *International Workshop on Recognition, Analysis, and Tracking of Faces and Gestures in Real-Time Systems*, pp. 70-76, Kerkyra, Greece, September 1999.

[3] Ji, Q. and Zhu, Z. Non-Intrusive Eye and Gaze Tracking For Natural Human Computer Interaction, *MMI-interaktiv Journal, Special Issue on Eye-Gaze Tracking for Human Computer Interaction*, 2003.

[4] Ji, Q. and Zhu, Z. Eye and Gaze Tracking for Interactive Graphic Display, *Machine Vision and Applications*, vol. 15, no. 3, pp. 139-148, 2004.

[5] Magee, J. J., Scott, M. R., Waber, B. N. and Betke, M. EyeKeys: A Real-Time Vision Interface Based on Gaze Detection from a Low-grade Video Camera, *Workshop on Real-Time Vision for Human-Computer Interaction (RTV4HCI)*,Washington D.C, July 2004.

[6] Matsumoto, Y. and Zelinsky, A. An Algorithm For Real-Time Stereo Vision Implementation of Head Pose and Gaze Direction Measurement, *Automatic Face and Gesture Recognition*, pp. 499-504, March 2000.

[7] Mimica, M.R.M., and Morimoto, C.H. A Computer Vision Framework For Gaze Tracking, *Computer Graphics and Image Processing (SIBGRAPI 2003)*, pp. 406 412, October 2003.

[8] Newman, R., Matsumo, Y., Rougeaux, S. and Zelinsky, A. Real-Time Stereo Tracking for Head Pose and Gaze Estimation, *Automatic Face and Gesture Recognition*, pp. 122-128, March 2000.

[9] Ngyun, K., Wagner, C., Koons, D., and Flickner, M. Differences in the Infrared Bright Pupil Response of Human Eyes, *Eye Tracking Research and Applications Symposium (ETRA 2002)*, pp. 133-138, 2002.

[10] Ohno,T., Mukawa, N., and Yoshikawa, A. Freegaze: A Gaze Tracking System forEveryday Gaze Interaction, *Eye Tracking Research and Applications Symposium (ETRA 2002)*, March 2002.

[11] Vezhnevets, V. and Degtiareva, A. Robust and Accurate Eye Contour Extraction. *Graphicon 2003*, pp. 81-84, Moscow, Russia, September 2003.

[12] Zhu, Z. and Yang, J. Subpixel Eye Gaze Tracking, *Automatic Face and Gesture Recognition*, p. 131, May 2002.