

A Method for Real-Time Eye Blink Detection and Its Application

Chinnawat Devahasdin Na Ayudhya
Mahidol Wittayanusorn School
Puttamonton, Nakornpatom 73170, Thailand
Chinnawat.Deva@gmail.com

Thitiwan Srinark
Department of Computer Engineering
Faculty of Engineering, Kasetsart University
Jatujak, Bangkok 10900, Thailand
Thitiwan.S@ku.ac.th

Abstract

Various human behaviors can be indicated by eye blink patterns. In this paper, we present a method based on image processing techniques for detecting human eye blinks and generating inter-eye-blink intervals. We applied Haar Cascade Classifier and Camshift algorithms for face tracking and consequently getting facial axis information. In addition, we applied an Adaptive Haar Cascade Classifier from a cascade of boosted classifiers based on Haar-like features using the relationship between the eyes and the facial axis for positioning the eyes. We proposed a new algorithm and a new measurement for eye blinking detection called “the eyelid’s state detecting (ESD) value.” The ESD value can then be used for examining the open and close states of eyelids. Our algorithm provides a 99.6% overall accuracy detection for eye blink detection. We generated inter-eye-blink interval graphs by differencing between two consecutive eye blink states. The graphs show that the common blinks of human presents short and long durations alternatively.

Key Words - face detection; eye detection; blink detection; inter-eye-blink interval.

1. Introduction

The goal of our research is to propose a new method to efficiently track eyes of a person from video image sequences; and propose a new algorithm to analyze the eyelid’s states or the ESD value. The analyzed states are then further used for generating an inter-eye-blink interval graph, which can be used to study different eye-related behavior analyses, e.g., fatigue test for drivers, sleep driving, physical-eye related diseases and lie detecting process.

Compared to eye-blink detection by using some head-mounted devices such as in [1] and some commercial eye trackers, eye-blink detection from video images may not be as accurate as them. However, this is usually compensated by greater ease of use, non-invasiveness and much lower cost. For our purpose, the eye tracking can run in real-time, without any additional hardware (like IR illumination for example) and be capable of operating under varying indoor conditions (typical office environment).

2. Previous work

There are many researches related with this work. For face detection, a chain of single-feature filters, Haar Cascade Classifier [2] is used for identifying sub-region image. With the fast calculation of integral image technique; it can work in real time. Camshift algorithm [3] or “Continuously Adaptive Mean Shift” uses a combination of colors represented as Hue from the HSV color model and skin probability for face tracking. The algorithm can track different types of facial views, not only the front view.

For eye detection, a cascade of boosted classifiers based on Haar-like features [4] is built by two training data sets, positive samples and negative samples. A learning algorithm, Adaptive Boost, is used to construct a strong classifier from the weak classifier. An improved method for eye extraction using deformable templates [5] was proposed. A new size term and an eye corner finder are introduced to prevent over-shrinking, improve speed and accuracy of fitting. C-BDA [6] is a biased discriminant analysis using the covariance of composite vectors. In the hybrid cascade detector constructed for eye detection, Haar-like features are used in the earlier stages and composite features obtained from C-BDA are used in the later stages.

For eye tracking, a strong tracking finite-difference extended Kalman filter algorithm, and overcome the modeling of nonlinear eye tracking was presented [7]. The filter uses finite-difference method to calculate partial derivatives of nonlinear functions to eye tracking.

For eye blink detection, open and close eye templates are used for blink pattern decisions based on correlation measurement. The method was specifically useful for people with severely paralyzed [8]. Normal flow and deterministic finite state machine (DFSM) with three states, steady state, opening state and closing state, use for calculating eye blink characteristics [9]. A real-time eye blinking detection was proposed based on SIFT feature tracking with GPU based implementation [10].

There was an interesting computer based study of eye blink in patients with moderately dry eye [11]. The study performed by evaluating the reflex picture of the cornea while each participant sat still with the head fixed in a dark room. The study concluded that typical blink patterns were found to be a relatively time-independent, irregular pattern. Their study suggested that typical patterns of eye blink

during conversation and VDT use might exist. Many patients showed alternating inter-blink periods of shorter and longer durations. Other patients showed initially shorter inter-eye-blink interval over 2–4 minutes followed by a relatively regular pattern of longer inter-eye-blink interval.

3. Algorithms

The eye blink analysis practically consists of three analyses based on the human face components: face detection, eye detection and eye blink detection.

3.1 Face detection

We applied Haar Cascade Classifier and Camshift algorithms for face detection and face tracking, respectively. The Camshift algorithm is more efficient for tracking than the Haar Cascade Classifier when working with multiple image frames, and it can track different types of facial views, not only the front view. The size and angle of the face location are adjusted each time when it shifts. The scale and orientation, which are best fit to the face-probability pixels inside the new location, are selected. As the result, we can know the ellipse estimation of each face, which is later used to approximate an axis of the eyes. Figure 1 shows example results of face detection and tracking.



Figure 1. Example results of Haar Cascade Classifier and Camshift algorithms

We then compute the smallest rectangular area, which can fully cover the estimated ellipse. This rectangular area is later used for eye detection. Figure 2 shows the relationship between an ellipse and its rectangular area. Suppose the center of the ellipse is at (C_x, C_y) . The top-bordered, the bottom-bordered, the left-bordered and the right-bordered linear equations are shown in (1), (2), (3) and (4), respectively.

$$y = C_y - 2\sqrt{\frac{AD}{4AC - B^2}} \quad (1)$$

$$y = C_y + 2\sqrt{\frac{AD}{4AC - B^2}} \quad (2)$$

$$x = C_x - 2\sqrt{\frac{CD}{4AC - B^2}} \quad (3)$$

$$x = C_x + 2\sqrt{\frac{CD}{4AC - B^2}} \quad (4)$$

Where

$$A = a^2 \cos^2 \theta + b^2 \sin^2 \theta$$

$$B = \sin 2\theta(a^2 - b^2)$$

$$C = a^2 \sin^2 \theta + b^2 \cos^2 \theta$$

$$D = a^2 b^2$$

Note that based on the image coordinates, the value of x increases from left to right, and the value of y increases from top to bottom.

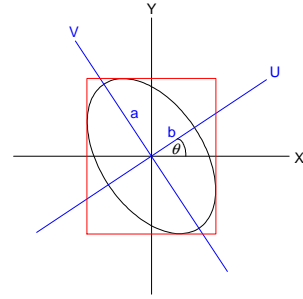


Figure 2. The face ellipse and the circumscribed rectangle

3.2 Eye detection

We used Adaptive Boost to train a cascade of boosted classifiers based on Haar-like features. Two training data sets, positive samples and negative samples, are required for this construction. Face databases [12, 13, 14] are used and all of these sample data were prepared by using *objectmarker* [4], which is easy for cropping desired areas by using a mouse. We used images of an eye with the eyebrow as the positive samples because there are more detectable details than using only an eye. We generated 3,327 positive image samples, and 6,478 negative image samples for training. Figure 3 shows our eye detection results.

However, from our experiments, we found that the cascade of boosted classifiers based on Haar-like features gives small accuracy rate. Therefore, we further developed Adaptive Haar Cascade Classifier by using the relationship between human's eyes and facial axes. Based on the fact that the eye axis, which is estimated by connecting the left and the right eyes' centers, is perpendicular with the major axis of the face ellipse; and both eyes should be symmetrical. Instead of using only the first two detected elements, which have highest recognition rates, from the Haar Cascade Classifier as the eyes, we used all detected elements from the Haar Cascade Classifier, as shown in Figure 4, and applied some geometric tests for checking which pair is probable the eyes.

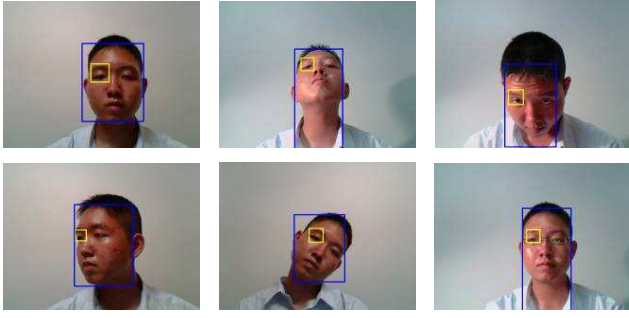


Figure 3. Example of eye detection results using the cascade of boosted classifiers based on Haar-like features

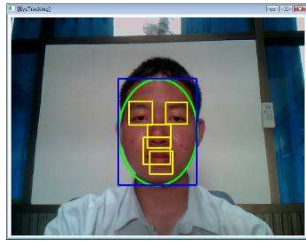


Figure 4. All detected elements from Haar Cascade Classifier

Our geometric tests include two conditions: (1) if the angle between the major facial axis and the potential eye axis is within 90 ± 10 degrees, and (2) if the sizes of the two potential rectangles that indicate two eyes are similar or the difference between their rectangular sizes is within 20%. The two rectangular areas are considered as the two eye area if they satisfy both conditions. However, if there is more than one rectangular pair under such conditions, we choose the pair with the highest position.

Note that there can be a case, which a facial view may be estimated as a horizontal ellipse as shown in Figure 5. Therefore, we need to additionally test which axis (either the major or minor facial axes) should be used to compare with the potential eye axis. Figure 6 shows a horizontal ellipse of a looking down face. The major axis lies at the angle α with the x-axis, and the minor axis lies at the angle θ with the x-axis. We test if $|\theta| > 60^\circ$, we use the minor facial axis instead of the major facial axis to compare with the potential eye axis.



Figure 5. A horizontal ellipse of a looking down face

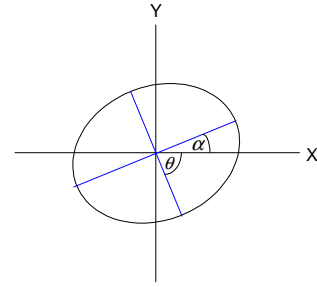


Figure 6. A horizontal ellipse and related angles

3.3 Blink Detection

We propose an ESD (Eyelid's State Detecting) value, which is a measurement used to classify the state of eyelid, open or close. The value can be computed by using the algorithm shown in Figure 7. The objective of the algorithm is to find the minimum threshold, which brings the binary image having at least one black pixel after applying median blur filtering. In this algorithm, we use a half-bottom eye image from the selected area by the previous algorithm. We then threshold the image with the threshold value (begin with 0). After that, we apply a median blur filter to the threshold image and check whether at least one black pixel appears. If there is no black pixel, we increase the threshold value and follow the same sequence, but if there is more than one black pixel, we terminate the process and get the ESD value as that threshold. For a faster computation, a binary search implementation is suggested.

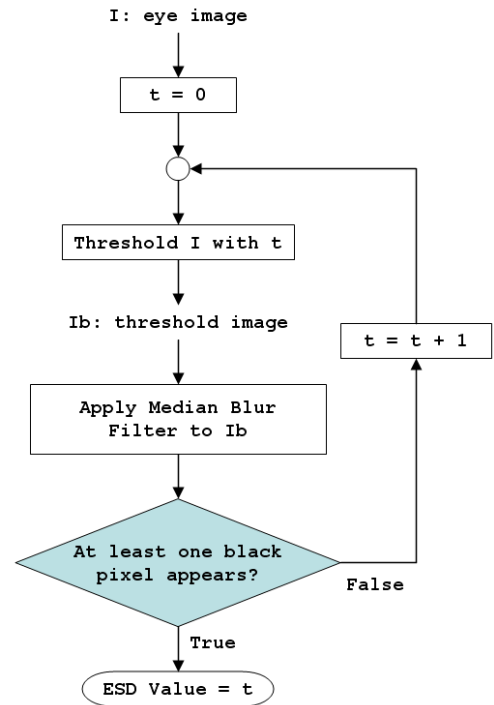


Figure 7. The proposed algorithm to find the ESD value

Figure 8 shows example images of two states of eyelid, open and close. The threshold images and the median blur images of the open state and the close state are shown in Figure 9 and Figure 10, respectively. The first row images present the threshold images and the second row presents the corresponding blur images. The ESD value of the presented open state is between 10 and 20, which is 18, and the ESD value of the presented close state is between 30 and 40, which is 36.

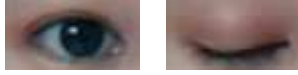


Figure 8. The open state and the close state of eyelid

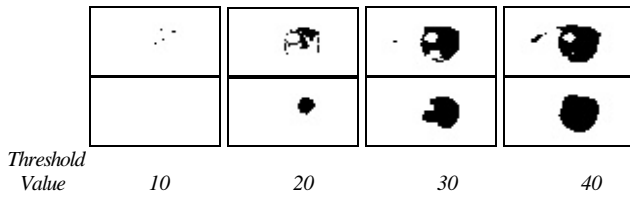


Figure 9. The threshold images (above) and the corresponding median blur images (below) of the open state

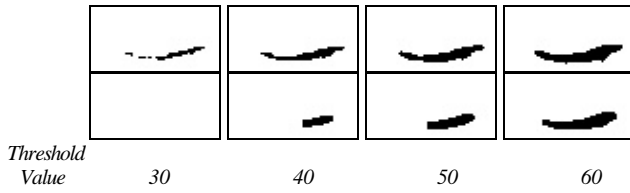


Figure 10. The threshold images (above) and the corresponding median blur images (below) of the close state

We illustrate an example of a graph of ESD values in Figure 11. The graph was produced by multiplying the ESD value of the left eye and the right eye from a part of our experimental video sequences. From the graph, it can be seen that there are some peak ESD values, which are marked by circles. These peak ESD values appear in the frames which the subject moves his/her eyelids or blinks. Therefore, we can use the ESD values to classify the state of eyelid such that the high ESD value indicates close state and the low ESD value indicates open state.

We defined a deterministic finite state machine operation to determine the eye blink characteristics as shown in Figure 12. The state machine consists of four states: steady state I (open), closing state or steady state II, opening state or steady state III, and blinking state. The decision variables for state changing include *slope*, *SPS*, and *SNS*, which are calculated based on the magnitudes of an ESD value graph. The values of *slope* are based on

slope values computed from an ESD value graph. The value of *SPS* (Sum of Positive Slope) is equal to sum of consecutive positive slope values. The value of *SNS* (Sum of Negative Slope) is equal to sum of consecutive negative slope values.

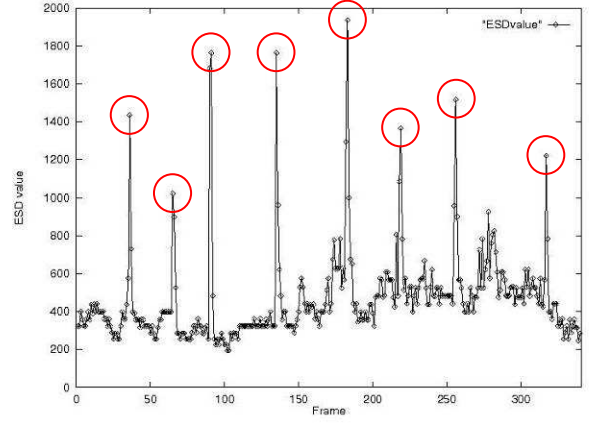


Figure 11. The ESD value across eye blinks and the real-closing eyelid frames (marked by circles)

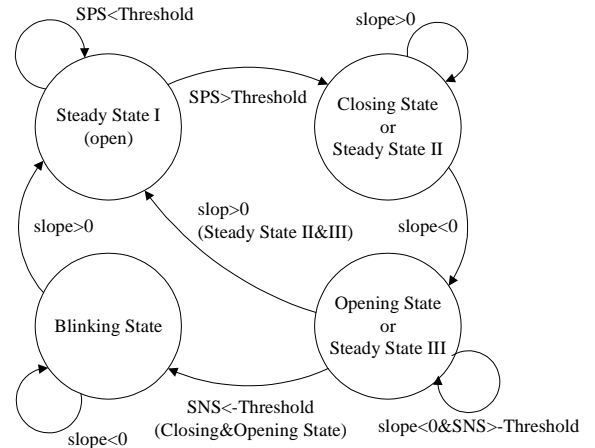


Figure 12. The deterministic finite state machine operation for detection of eye blinks

Normally the process begins at the steady state I. The current state is changed to the closing state if *SPS* is larger than a threshold. At the closing state, we consider the value of *slope*; if *slope* is less than zero, the current state is changed to the opening state. At the opening state, we consider the values of *slope* and *SNS*. If *slope* is greater than zero, the current state is changed to the steady state I. If *SNS* is less than a negative threshold, the current state is changed to the blinking state, which means an eye blink is detected at this frame with an appropriate threshold.

4. Experimental Results

We implemented all of the algorithms in C++ by using Bloodshed Dev-C++ 4.9.9.2 as a compiler, and OpenCV [4, 15, 16, 17, 18] as an image processing and computer vision library. GNUplot is used for graph generating. We run our program in a Dell Inspiron notebook with Intel(R) Core(TM)2 Duo CPU T7250 at 2.00 GHz and 2.00 GB RAM. The video camera is a built-in webcam with the resolution at 320×240 (2.0 Megapixel) and the frame rate at 30 frames per second. Note that OpenCV can operate at 7 frames per second by average.

For eye detection, the efficiencies of the Haar Cascade Classifier and the Adaptive Haar Cascade Classifier were tested on 1,000 daily life photos provided by Huang *et.al* [19]. Note that, every image is 100 percent face detection.

We show the accuracy rates of eye detection methods, Haar Cascade Classifier and adaptive Haar cascade classifier in Table 1. The experiment shows that Adaptive Haar Cascade Classifier method is more efficient than the Haar Cascade Classifier method with 22.7% of accuracy improvement.

For blink detection, ESD Values are evaluated on our video records. Consecutive frames were chosen randomly from each record. The results are shown in Table 2.

Table 1. The accuracy results of the two eye detection methods

	Haar Cascade Classifier	Adaptive Haar Cascade Classifier
Accuracy (%)	59.7	82.4

Table 2. The result of blink detection

# Frames	Blink (True)	Not Blink (False)
Detected as Blink (Positive)	1944 (TP)	195 (FP)
Detected as Not Blink (Negative)	156 (FN)	84905 (TN)

We compute the overall detection accuracy and the detection accuracy of the eye blink detection by using (5) and (6), respectively. Where *TP* is the number of frames that are correctly detected eye blinks (true positive); *FN* is the number of frames that show eye blinks but the program is not detected (false negative); *FP* is the number of frames that are reported as eye blinks but they are not (false positive); and *TN* is the number of frames that are correctly reported as no blinks (true negative).

$$Overall = \frac{TP + TN}{TP + FP + FN + TN} \times 100\% \quad (5)$$

$$Detection = \frac{TP}{TP + FN} \times 100\% \quad (6)$$

Therefore, our overall detection accuracy is 99.6%, and our detection accuracy is 92.6%. From the experiments, inaccuracy of our eye blink detection is occasionally occurred in two situations. The first situation is when a subject moves his/her head swiftly. In this situation, even though we can correctly figure out the eye positions, we cannot correctly determine eye blink states. This is because eye images are blurry such that skin colors blend with the colors of the eye areas. The second situation is when a subject changes the eye focus to the lower area so eyelids partially close and sometime the subject may bow his/her head as well (see an example in Figure 5).

We show the average execution time of a single frame by using our algorithms in Table 3, where we performed experiments on four video data files; and each video contains 1,000 frames. It can be seen that Haar Cascade Classifier takes the longest execution time. However, we only need it for the first frame analysis, or when we lose track of eyes. The other three algorithms are used in every frame computation. The total average execution of these three algorithms is only 15.787 ms. Therefore, our method can work in real time.

Table 3. The result of blink detection

Algorithms	Time (ms)
Haar Cascade Classifier (first time face detection)	23.280
Camshift (face tracking)	0.020
Adaptive Haar Cascade Classifier (eye detection)	14.619
ESD value calculation	1.148

In an application, we recorded video files and used our method to automatically detect eye blinks of four volunteers while they were using a computer for 45 minutes. The inter-eye-blink interval is calculated from a different frame number between two consecutive blinking frame *i* and *i* + 1. Figure 13 shows the inter-eye-blink interval graph of each volunteer.

The graphs show that typical blink of human contains alternating inter-eye-blink periods of shorter and longer durations. Where the results are similar to the results in patients with moderately dry eyes that the inter-eye-blink interval results fluctuate between period of shorter and longer durations, but nobody showed initially shorter and finally longer inter-eye-blink interval [11].

5. Conclusions

We developed the Adaptive Haar Cascade Classifier to increase the efficiency of the Haar Cascade Classifier. It provides 22.7% of accuracy improvement for eye detection. Calculated from our new method, ESD Value can classify the state of eyelid, open or close. It provides a 99.6% overall detection accuracy, and 92.6% detection accuracy. Our method takes only 15.787 ms as average execution time for each frame, therefore, it can work efficiently in real-time applications. According to the

study, the result of graph analysis on four sets of 45-minute samples show that the typical blink of human contains alternating inter-blink periods of shorter and longer durations.

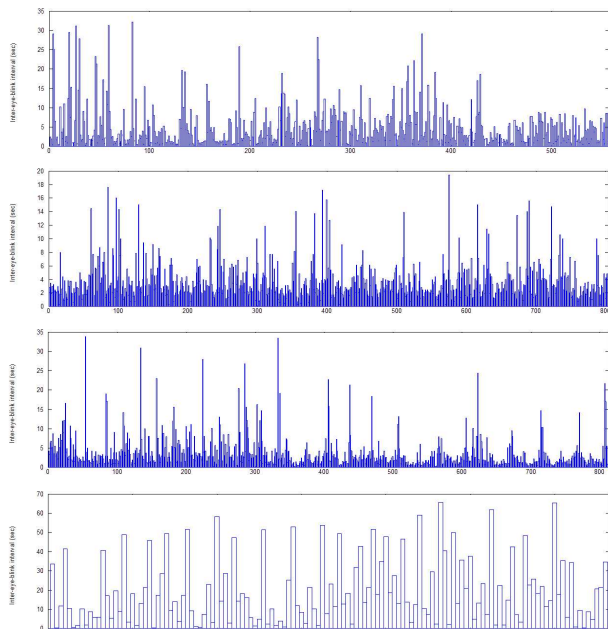


Figure 13. The inter-eye-blink interval graphs of four volunteers

6. Acknowledgements

This research was supported by JSTP (Junior Science Talent Project), YSC (Young Scientist Competition), Associate Professor Doctor Anuchit Poonyathalang, Department of Ophthalmology, Faculty of Medicine, Ramathibodi Hospital, Patoomsiri Songsiri, Department of Computer & Technology, Mahidol Wittayanusorn School, and Department of Computer Engineering, Faculty of Engineering, Kasetsart University.

7. References

- [1] Topal, C., Gerek, Ö. N. and Doğan, A., "A head-mounted sensor-based eye tracking device: eye touch system," *Proc. of the 2008 Symp. on Eye tracking research & applications*, 2008, pp. 87-90.
- [2] Viola, P. and Jones, M., "Rapid Object Detection using a Boosted Cascade of Simple Features," *Proc. of the Conf. on Computer Vision and Pattern Recognition (CVPR)*, Hawaii, USA, December 9-14, 2001, Vol. 1, pp. 511-518.
- [3] Bradski, G. R., "Computer Video Face Tracking for Use in a Perceptual User Interface," *Intel Technology J.*, Q. 2, 1998.
- [4] Adolf, F., "How-to build a cascade of boosted classifiers based on Haar-like features," *OpenCV's Rapid Object Detection*, 2003.
- [5] Kuo, P. and Hannah, J., "An Improved Eye Feature Extraction Algorithm Based on Deformable Templates," *Proc. of the IEEE Int. Conf. on Image Processing (ICIP)*, Genoa, Italy, September 11-14, 2005, pp. 1206-1209.
- [6] Kim, C. and Turk, M., "Biased Discriminant Analysis Using Composite Vectors for Eye Detection," *Proc. of the 8th IEEE Int. Conf. on Automatic Face and Gesture Recognition*, Amsterdam, The Netherlands, September 17-19, 2008.
- [7] Zhang, J. and Zhang, Z., "Application of a Strong Tracking Finite-Difference Extended Kalman Filter to Eye Tracking," *Proc. of the Int. Conf. on Intelligent Computing (ICIC)*, Kunming, China, August 16-19, 2006, pp. 1170-1179.
- [8] Grauman, K., James, Betke, M., Gips, J. and Bradski, G. R., "Communication via Eye Blinks – Detection and Duration Analysis in Real Time," *Proc. of the IEEE Computer Society Conf. on Computer Vision and Pattern Recognition (CVPR)*, Hawaii, USA, December 9-14, 2001, Vol. 1, pp. 1010-1017.
- [9] Heishman, R. and Duric, Z., "Using Image Flow to Detect Eye Blinks in Color Videos," *Proc. of the IEEE Workshop on Applications of Computer Vision (WACV)*, Austin, Texas, USA, February, 2007.
- [10] Lalonde, M., Byrns, D., Gagnon, L., Teasdale, N. and Laurendeau, D., "Real-time eye blink detection with GPU-based SIFT tracking," *Proc. of the 4th Canadian Conf. on Computer and Robot Vision (CRV)*, Montreal, Quebec, May 28-30, 2007, pp. 481-487.
- [11] Schlote, T., Kadner, G. and Freudenthaler, N., "Marked reduction and distinct patterns of eye blinking in patients with moderately dry eyes during video display terminal use", *Graefe's Archive for Clinical and Experimental Ophthalmology*, Vol. 242, No. 4, April, 2004, pp. 306-312.
- [12] Tarr, M. J., "Face-Place," *Face-Place-tarrlab*, Department of Cognitive and Linguistic Sciences, the Brown Center for Vision Research, the Brown Institute for Brain Science, Brown University, <<http://titan.cog.brown.edu:8080/TarrLab/face-place>> (9 August 2007).
- [13] Stegmann, M. B., "The IMM Face Database," *Free data sets for statistical models of shape*, Technical University of Denmark, <<http://www2.imm.dtu.dk/~aam>> (14 September 2007).
- [14] Georgiades, A.S., Belhumeur, P.N. and Kriegman, D.J., "From Few to Many: Illumination Cone Models for Face Recognition under Variable Lighting and Pose," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. 23, No. 6, pp. 643-660.
- [15] Hewitt, R., "Seeing with OpenCV A Computer-Vision Library," *SERVO*, January 2007, pp. 62-66.
- [16] Hewitt, R., "Seeing with OpenCV Finding Faces in Images," *SERVO*, February 2007, pp. 48-52.
- [17] Hewitt, R., "Seeing with OpenCV Follow That Face," *SERVO*, March 2007, pp. 36-40.
- [18] Pisarevsky, V., "OpenCV Object Detection: Theory and Practice," *Slide Presentation*, Intel Corporation, Software and Solutions Group, June 10, 2007.
- [19] Huang, G. B., Ramesh, M., Berg, T. and Learned-Miller, E., "Labeled Faces in the Wild: A Database for Studying Face Recognition in Unconstrained Environments," *Technical Report 07-49*, University of Massachusetts, Amherst, October, 2007.