In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Lihue, HI, December 2001.

Communication via Eye Blinks – Detection and Duration Analysis in Real Time

Kristen Grauman* Margrit Betke James Gips Gary R. Bradski

Vision Interface Group, MIT AI Lab

Image & Video Computing, Boston University

EagleEyes, Boston College

Visual Interactivity Group, Intel Corporation

Abstract

A method for a real-time vision system that automatically detects a user's eye blinks and accurately measures their durations is introduced. The system is intended to provide an alternate input modality to allow people with severe disabilities to access a computer. Voluntary long blinks trigger mouse clicks, while involuntary short blinks are ignored. The system enables communication using "blink patterns:" sequences of long and short blinks which are interpreted as semiotic messages. The location of the eyes is determined automatically through the motion of the user's initial blinks. Subsequently, the eye is tracked by correlation across time, and appearance changes are automatically analyzed in order to classify the eye as either open or closed at each frame. No manual initialization, special lighting, or prior face detection is required. The system has been tested with interactive games and a spelling program. Results demonstrate overall detection accuracy of 95.6% and an average rate of 28 frames per second.

1 Introduction

In recent years, there has been an effort to augment traditional human-computer interfaces like the keyboard and mouse with intelligent interfaces that allow users to interact with the computer more naturally and effectively. The goal is to develop computer vision systems that make computers perceptive to a user's natural communicative cues such as gestures, facial expressions, and gaze direction. Such systems are especially relevant for people who cannot use the keyboard or mouse due to severe disabilities.

The traditional human-computer interfaces demand good manual dexterity and refined motor control which may be absent or unpredictable for people with certain disabilities. The primary motivation of our research is to provide an alternative communication tool for people whose motor abilities are extremely limited by conditions ranging from cerebral palsy and traumatic brain injuries to multiple sclerosis or ALS. The access to information and enhanced communication that assistive technology provides is both practical and empowering for individuals with disabilities.

We propose a robust, accurate algorithm to detect eye blinks, measure their duration, and interpret them in real time to control a non-intrusive interface for computer users with severe disabilities. The method presented employs visual information about the motion of eyelids during a blink and the changing appearance of the eye throughout a blink in order to detect the blink's location and duration. Moreover, the system is designed to initialize itself automatically, and it adjusts for changes in the user's position in depth. Using the "Blink Link," as our prototype system is called, a user who is capable of blinking voluntarily can generate mouse clicks through his or her eye blinks in order to operate software applications requiring such input.

The system uses various computer vision techniques in combination. Eye blink motion is used to automatically locate the user's eyes in the video sequence. In particular, candidate motion patterns are compared against a stored model of the properties of actual eye blink motion in order to eliminate motion that is unlikely to have resulted from blinks. The location information gained from the blink motion then offers an opportunity to select an eye template online for further tracking. The correlation between the open eye template and the current eye in the scene reveals the extent of the eye's openness which, together with the complementary motion information obtained from both eye areas, allows us to classify the eye as either open or closed at each frame.

An array of techniques have been explored previously for locating eyes in images and eye blink detection. Methods for detecting the eyes include the use of

*Previously at Boston College. Email: kgrauman@ai.mit.edu. The work has been supported by NSF grants 9871219 and IIS-0093367.

ISBN 0-7695-1272-0/01 $10.00 (C) 2001 IEEE

gradient flow fields [11], color-based techniques for detection of the eye sclera [2], horizontal gradient maps of a skin-colored region [16, 17], and pupil detection using infrared lighting [9, 12, 19]. Temporal differencing is often used to segment moving regions of interest from a stable background [3, 4]. Methods for analyzing the eye and its closure motion are suggested in Refs. [1, 3, 13, 14, 16, 18]. A blink detector has been developed to detect drowsy drivers [13]. Preliminary work on facial feature tracking has been reported that provides video-based interfaces for people with disabilities [6, 15]. We are not aware of any papers that address the issues of communication interfaces which operate on eye blinks. Such interfaces demand the robust and accurate classification of voluntary and involuntary blinks, must work with assistive technology software, and require exceptionally fast processing.

Our contribution is to provide a real-time system which consistently runs at 27–29 frames per second (fps), is completely non-intrusive and requires no manual initialization, prior face detection, or special lighting. The system can reliably classify blinks as voluntary or involuntary based on their duration. Thus, it is found to be a reasonable communication interface for users who have the ability to blink their eyes. Our communication system has also been tested for recognizing substantial deformations of other features.

Alternative communication systems for disabled computer users include head-mounted tracking devices, tongue or hand activated switches, sip-and-puff mouth-controlled joysticks, camera-based and electro-oculographic gaze estimators [5, 6, 15]. The goal of introducing eye blink detection functionality in a camera-based system is to provide another point of access for those users who may not be capable of motor controls that some of the above methods demand.

User testing for the Blink Link is being performed at Boston College's Campus School, a school for children with various severe disabilities. Currently, children there use two systems as mouse replacements: the "Camera Mouse" system uses a video camera to perform facial feature tracking [6], and the "Eagle Eyes" system measures the user's electro-oculographic potential to estimate gaze direction [5, 7]. Children use the systems to spell out messages, play games, and even participate in distance learning programs on the web.

2 System Overview

The system design can be broken down into four steps as shown in Fig. 1: (1) motion analysis for the purpose of locating the eyes, (2) eye tracking, (3) blink detection and length measurement, and (4) interpretation. The eyes are located automatically by considering motion information between two consecutive frames and determining if this motion is likely to be caused by a blink. Once found in this manner, a grayscale template is extracted from the blink location of one eye. The eye is tracked and constantly monitored to establish to what extent it is open or closed at each frame. A blink's duration is defined as the count of consecutive frames of closure. If at any time the eye tracker is believed to be lost, then it is re-initialized by repeating motion analysis on the subsequent involuntary blinks.

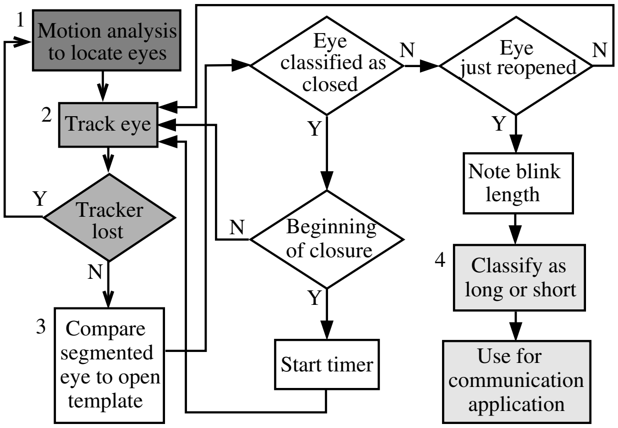

1
Motion analysis to locate eyes
2
Track eye
Y
Tracker lost
N
3
Compare segmented eye to open template
Eye classified as closed
N
Y
N
Beginning of closure
Y
Start timer
Eye just reopened
N
Y
Note blink length
4
Classify as long or short
Use for communication application

Figure 1: System overview.

2.1 Motion Analysis

During the first stage of processing, the eyes are automatically located by searching temporally for "blink-like" motion. This method analyzes a sequence of the user's involuntary blinks and exploits the redundancy provided by the fact that humans naturally blink regularly. The bi-directional difference image $[D]_{i,j} = |([F_t]_{i,j} - [F_{t-1}]_{i,j})|$ is formed from previous frame image F_{t-1} and current frame image F_t for all pixels (i, j) in order to capture both increasing and decreasing brightness changes. The difference image is thresholded to produce a binary image representing regions of significant change, i.e. motion, in the scene.

Next the image undergoes erosion with a cross-shaped convolution kernel in order to eliminate spurious pixels generated by phenomena such as flickering lights, high-contrast edges, or arbitrary jitter. For example, the sharp contrast along the edge between the face and the hair or shadow on the neck permits only a negligible amount of movement to result in a significant brightness change. Such irrelevant motion is noise to the system, and therefore removed by the erosion process (see Fig. 2).

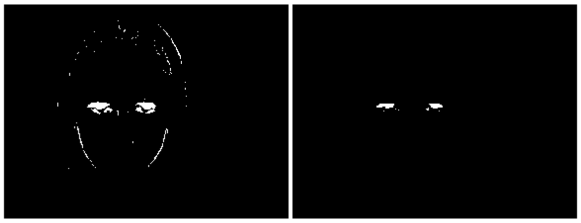


Figure 2: Thresholded difference image prior to erosion (left), and same image after erosion (right). Erosion removes noise caused by insignificant motion in the scene.

Finally, candidate eye ”blobs” are extracted by labeling the connected components in the pre-processed difference image. Each possible pairing of the components is analyzed to determine if the pair is likely to represent blink motion.

Each candidate component pair has a vector of properties $\mathbf{p} = [s_x, s_y, w_r, w_l, h_r, h_l]$ where s_x, s_y are the distances in x and y between each respective component's centroid, and w_l, w_r, h_l, and h_r denote the width and height of each component, normalized by their separation from one another. The candidate pairs first undergo several filters that eliminate pairs whose properties make them anthropomorphically infeasible, such as excessive separation between the components in the y-axis, or components whose dimensions are disproportional to their separation from one another. Large samples comparing the properties of non-blink motion component pairs to those of true blink motion pairs revealed several clear distinctions between the classes. As a result, the majority of candidate motion components can be quickly discarded by the filters to avoid consuming additional online computation resources (see Figs. 3 and 4).

Subsequently, surviving candidate pairs are compared to a model of known blink-pair measurements by calculating the weighted Mahalanobis distance d between the candidate pair's vector of properties $\mathbf{p}$ and the mean vector of blink-pair properties $\boldsymbol{\mu}$, where

$$d^2 = (\mathbf{p} - \boldsymbol{\mu})^t \Sigma^{-1} (\mathbf{p} - \boldsymbol{\mu}). \qquad (1)$$

The mean vector $\boldsymbol{\mu}$ and covariance matrix Σ for computing distance d are produced by manually identified blink-pairs at different depths and face orientations during training. Pairs having distances less than a threshold are classified as non-members (not blinks), and those classified as members proceed as candidates. The Mahalanobis distance measure was chosen because of its sensitivity to inter-variable changes in the training data.

For a given frame, if there exists a pair of motion components whose Mahalanobis distance is less than the threshold, then these components are the blink candidates for that frame. If there happens to be more than one component pair that survives the threshold, then only the pair with the lowest Mahalanobis distance is considered. Believing that the motion of the candidate was caused by the quick closing and opening of the eyes, a template of the open eye may be captured instants (frames) later from the location in the image of one of the eye components. The template's size is based on the bounding box of the segmented motion blob. The area of segmented motion is directly proportional to the size of the eye that caused it. Therefore, the automatically chosen templates are depth-sensitive and accurately proportional in size to the user's eye at the time of initialization.

During the initialization phase, n templates resulting from the n best candidate pairs are collected in this

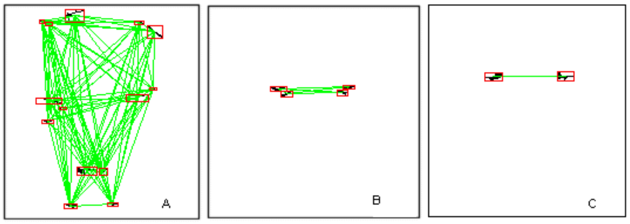


Figure 3: Thresholded, segmented difference image showing arbitrary motion (A), two candidate pairs falling below the Mahalanobis distance threshold (B), and one candidate pair identified as a blink (C). Red boxes bound regions of motion and green lines connecting the boxes indicate component pairings.

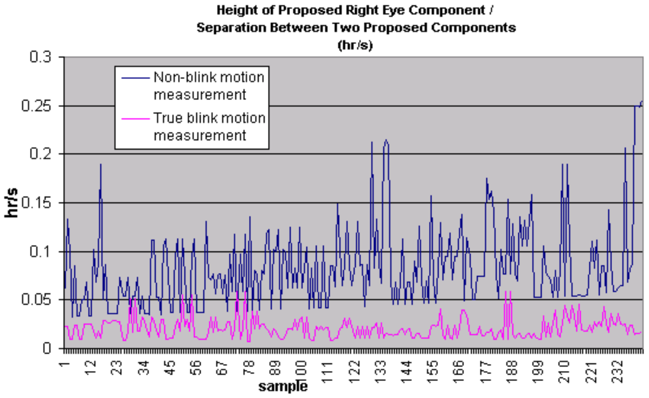


Figure 4: Example of a filter applied to candidate eye component pairs. Values associated with instances of true eye blinks are significantly lower than those of non-blink motion in the scene. A filter can therefore be used to quickly discard candidate motion blobs that do not describe eye blinks.

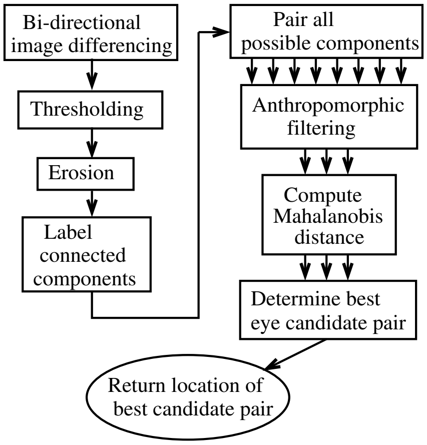

Bi-directional image differencing
Thresholding
Erosion
Label connected components
Pair all possible components
Anthropomorphic filtering
Compute Mahalanobis distance
Determine best eye candidate pair
Return location of best candidate pair

Figure 5: Details of motion analysis phase.

manner. Finally, the system determines which open eye template is used by comparing all n choices against a stored model of the open eye and selecting the template with the highest correlation score.

2.2 Eye Tracking

Motion analysis alone is not sufficient to give the highly accurate blink information desired. It does not provide precise duration information, and multiple component pair candidates may occur sequentially as the result of a single blink. Relying on motion would make the system extremely intolerant of extra motion due to facial expressions, head movement, or gestures. The user must be allowed to move his or her head with relative freedom if necessary.

Following initial localization, a fast eye tracking procedure maintains exact knowledge about the eye's appearance. Thus, the eye may be evaluated for amount of closure at the next stage. As described, the initial blink detection via motion analysis provides very precise information about the eyes' positions. Consequently, a simple tracking algorithm suffices to update the region of interest centered around the eye.

The system utilizes the normalized correlation coefficient $R(x,y) =$

$$\frac{\sum_{y'=0}^{h}\sum_{x'=0}^{w}\mathbf{T}(x',y')\mathbf{I}(x+x',y+y')}{\sqrt{\sum_{y'=0}^{h}\sum_{x'=0}^{w}\mathbf{T}(x',y')^2\sum_{y'=0}^{h}\sum_{x'=0}^{w}\mathbf{I}(x+x',y+y')^2}}$$

where $\mathbf{T}(x',y') = T(x',y') - \bar{T}$, $\mathbf{I}(x+x',y+y') = I(x+x',y+y') - \bar{I}(x,y)$, and $T(x,y)$ and $I(x,y)$ are the brightness of the pixels at (x,y) in the template and source image, respectively, and $\bar{T}$ is the average value of the pixels in the template raster and $\bar{I}(x,y)$ is the average value of the pixels in the current search window of the image. The coefficient $R(x,y)$ is a measure of match between the open eye template and all points within the small search region surrounding the location of the eye given from the previous frame. In this way, the current eye position is updated nearly thirty times per second and remains accurate sparing dramatic, sudden head movements or significant changes in depth. For these events, it is critical that the tracker declare itself lost and re-initialize using blink motion analysis as discussed above. The tracker is believed to be lost if the best match score found using the correlation coefficient falls below a set threshold $F = 0.55$. The tracker does not get lost during the blink because the closed eye and its closely neighboring pixels bear enough similarity to the open eye template to pass the threshold.

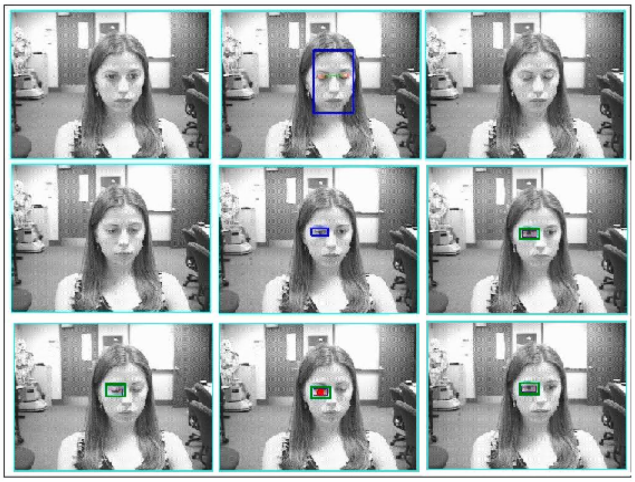


Figure 6: Intermittent frames from a sequence during the motion analysis phase when the template is being found automatically by the user's first several natural blinks. Rectangles around the face indicate that blink-like motion was detected. The small rectangle that appears around the eye three frames later indicates where the open eye template is being selected. The subsequent small rectangles indicate eye tracking. A red circle on top of the eye (third row, second column) indicates that a blink is believed to have just ended.

2.3 Blink Detection and Duration of Closure Measurement

As the eye closes, it begins to look less and less like an open eye; likewise, it regains its similarity to the open eye slowly as it reopens. This is a simple but powerful observation. During an eye blink, the best correlation scores reported by the tracker can be plotted across

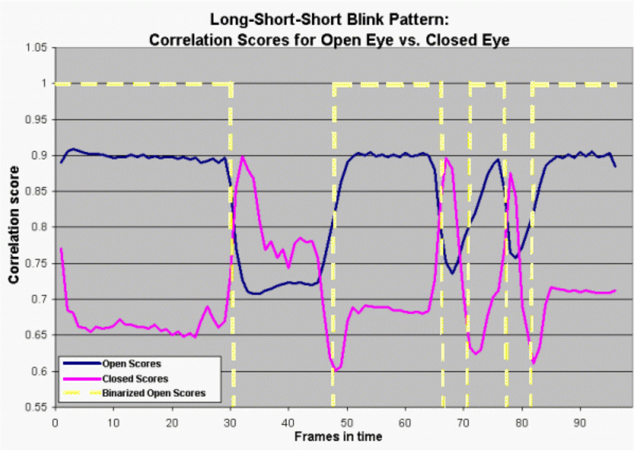


Figure 7: Correlation scores over time comparing the user's eye at each frame to both the open eye template and the closed eye template. The open eye scores present a waveform indicating the captured blink pattern: *long, short, short*. Such samples were collected and used to identify an effective threshold O for classifying eyes as opened or closed at each frame.

time to depict a clear waveform that illustrates the duration of successive blinks (see Fig. 7).

Experiments with the correlation scores between the actual eye and its closed template confirmed that this method succeeds from both aspects. However, the apparent correspondence of the two measures would make it redundant to compute both online, and so only the open eye correlation is used in the current system. Likewise, processing time may be conserved by tracking and computing the correlation for only one eye. The motion analysis above can be used to verify or refute the correlation score's findings. Since the motion components account for both eyes, correlating for the second eye would be superfluous and is therefore omitted. It is a simple task to specify in the software that a particular eye or both eyes be considered.

The waveforms representing degree of closure are so distinct that it is reasonable and useful to "binarize" the open correlation figures and thus classify the eye as open or closed at each frame. In addition to the threshold F that indicates the tracker is lost, a threshold $O = 0.85$ is needed for the minimum correlation score interpreted as an open eye. These two thresholds together allow the system to classify the eyes as being open, closed, or unfound at every single frame. In this way measuring blink length is possible. The system interprets only longer, voluntary blinks as meaningful; quick, involuntary blinks do not trigger mouse clicks. Analysis of video sequences from various users indicates that open eyes result in correlation scores ranging from 0.85 to 1.0, closed eyes result in correlation scores between 0.55 and 0.8, while "non-eye" segments of the facial region result in scores ranging from 0 to 0.4.

3 Communication Strategies Using Visual Cues

In feature-tracking systems such as the Camera Mouse [6], some small section of the face is tracked and is used to generate corresponding mouse motion. A user makes a selection or issues a mouse click by dwelling in the desired screen area for a given amount of time. Although the dwelling approach is generally effectual, it may result in undesired clicks being registered when a user needs to rest his or her head for a moment. The "Midas Touch" problem occurs when a user is unable to look anywhere without triggering some system response [10]. The blink method proposed in this work may be used in conjunction with such feature-tracking methods in order to provide a more active means of making selections. A prolonged blink is a more emphatic way to indicate voluntary selections.

The precise knowledge of blink duration offers an opportunity for a different communication strategy requiring only eye blinks: message encoding by blink patterns. At first glance, one might consider the application of the long/short inputs as a sort of Morse code in which any desired message is spelled letter by letter. Perhaps for some users with certain skills this is a feasible approach. However, a less demanding protocol was developed for this system. Long and short blinks are translated into a binary Huffman code where each prefix-free symbol is representative of a word or phrase in a certain small subset of related vocabulary. In practice, an individual controlling the computer with only blinks would need to select a vocabulary subset through some scanning software, and then proceed to link words or phrases into the desired message. Compared to straight Morse code, this approach requires fewer blink inputs and thus offers faster service.

The idea of losing and regaining correlation to a model in appearance was also extended to other facial features to provide additional functional inputs from body parts. For example, placing the tracker on the mouth while it is in a resting position allows a user to generate mouse clicks analogously to the blink method by simply opening and reclosing his or her mouth. Given the broad range of differing abilities possessed by users of assistive technology, multiple options for usage are certainly valuable.

4 Hardware

Development and testing of the system was done with a Sony EVI-D30 color video CCD camera, a Matrox Meteor II image capture board, and a 1 GHz dual processor PC with 256 MB RAM. Testing with people with disabilities was done on a 800 MHz single processor PC with 128 MB RAM. Grayscale images are processed at 320 × 240 pixels. When possible, two monitors are helpful but not necessary when running the Blink Link, for this allows one to monitor the status of the eye tracker.

5 Experiments and Discussion

The system has been tested for its accuracy as well as its usability as an input device. The following results are based on sessions with 15 different subjects without disabilities. They are summarized in Table 1. Several videos of our experiments are provided on the web [8].

In order to measure the accuracy of eye blink detection, video sequences were captured of each user sitting between two and four feet from the camera. The users were asked to blink naturally but frequently and exhibit mild head movement. Each sequence was processed by the system in real time. Processed images were saved and manually examined offline to determine precisely how the system had interpreted the data.

The results show that 98% of the blinks were detected. Only four blinks were missed out of the 204 actual blinks in the sequences. False positives were encountered five times, making the overall detection accuracy 95.6%.

Beyond simple detection accuracy, it was important to test the system's ability to classify blinks as involuntary (short) or voluntary (long). To achieve this, each subject was asked to blink out designated blink patterns. These sequences were then processed as above, where the ground truth was fixed to be the intended blink pattern. Patterns tested include sequences such as *long-short-short* or *short-long-short*. No parameters were altered for any single test. While the system allows a user to adjust the threshold on the minimum length of voluntary blinks online, part of the objective of these tests was to determine how well a single threshold would hold for multiple users given no instruction about what the system defines to be a "long" or "short" blink. The experiments show that a single threshold can be used and thereby reliably distinguish involuntary blinks across our users.

The system correctly detected all but two of the combined long and short blinks, yielding a 98% rate of accuracy for detection for these samples. 93% of the blinks were correctly classified as either long or short. The five misclassified blinks can most often be attributed to users who tend to make their long blinks virtually identical to their short blinks.

Table 1: Summary of Results	
Overall detector accuracy	95.6 %
Long/short classification accuracy	93.0 %
Usability score as an input device	93.6 %
Average frame rate	28 fps

In addition to the accuracy tests described above, experiments were also performed to study how feasible eye blinking is as an input modality for the computer. The idea is to use blinks to generate mouse clicks. Short, involuntary blinks are filtered out and only long, voluntary blinks cause a click. Applications used to test the blink input require no mouse movement; they operate entirely on mouse clicks regardless of the mouse location. While the eye tracking information may be used to generate mouse movement, for this system cursor motion is not included since users with severe disabilities do not necessarily have significant head or eye motion control.

The subjects were observed trying several simple games and one spelling program using the Blink Link. The games are commercial software intended as educational exercises for children with disabilities who can access a computer with a "switch," or a single input signal that the user triggers in some way. Here, the switch is the click that is generated by a blink. Because no cursor movement is considered, these games use a scanning mechanism in which the user is presented with one option at a time. The user must then blink a long blink when the desired option is presented. For example, in one game used to assess reflexes and coordination, a frog is depicted sitting in the marsh, waiting for flies to come by. The user must blink voluntarily when a fly appears in order to have the frog catch it with its tongue. In another game, images of familiar objects are shown, and the user must blink when the image of a matching object is shown (see Fig. 8).

The scores received when playing such games are good indicators of how well the system functions as an input device. Subjects played one round each of three different games. If a user's score is defined as the number of correct hits divided by the total sum of hits and misses, then the mean score recorded for the test subjects was 90%. Grand totals for the games played amount to 421 hits and 29 misses, making a cumulative score of 93.6%. Misses can be attributed to instances when the tracker was lost because of fast head movement, input blinks that were not long enough to meet the voluntary length threshold, or false positives

ISBN 0-7695-1272-0/01 $10.00 (C) 2001 IEEE

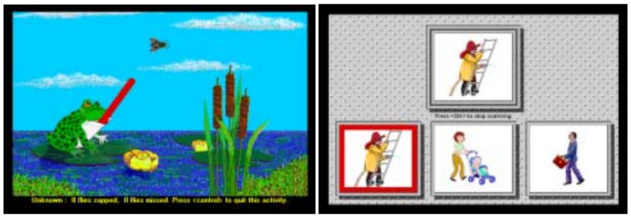


Figure 8: Sample games testing reaction time (left) and visual matching abilities (right). The red outlining box cycles through the options, and the user blinks when the matching image is outlined. Figure courtesy of Simtech Publications.

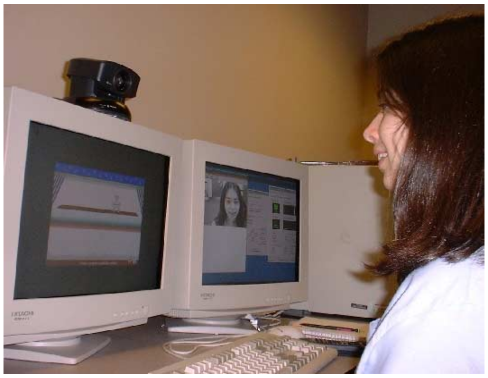


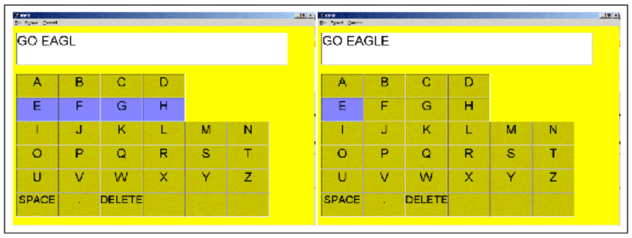
Figure 9: A user plays a scanning arcade game where the object is to throw a baseball at the plates on the shelf when they are outlined with a square. A long blink causes a ball to be thrown.

caused by involuntary blinks that should have been filtered out.

Users also tested a scanning spelling program using eye blinks. The program organizes the letters of the alphabet into groups and scans through these groups in order, line by line. The user waits for his or her desired row to be highlighted and then blinks. Next the program scans through each individual letter in that group, and the user blinks again when the desired letter is highlighted. The subjects were told to spell "GO EAGLES." The average time required to complete the task in one trial was 95 seconds (see Fig. 10). Users gain speed as they get accustomed to the system and accumulate practice.

The subjects also tried using their mouths to generate mouse clicks. In the current system, mouth control requires manual initialization of the template. It then works in a similar manner to the eye blink control. A brief opening and closing of the mouth generates a mouse click. For some people this is an easier motion to control, and thus a better input method.

The system is equipped to handle head movement,



GO EAGL

A	B	C	D		
E	F	G	H		
I	J	K	L	M	N
O	P	Q	R	S	T
U	V	W	X	Y	Z
SPACE	.	DELETE			

GO EAGLE

A	B	C	D		
E	F	G	H		
I	J	K	L	M	N
O	P	Q	R	S	T
U	V	W	X	Y	Z
SPACE	.	DELETE			

Figure 10: The scanning spelling game highlights each consecutive row of letters. A blink causes it to highlight each consecutive letter in that row, and another blink causes the letter to be printed. Here, the second row is highlighted in the left image. A blink selects the letters E–H to be scanned from left to right; another blink selects the letter E.

rotation in the image plane, and as much horizontal head turning or vertical nodding such that neither eye is completely occluded. Should the tracker become lost because of a sudden acceleration of the head, it is re-initialized within moments through blink motion analysis. Both eyes must therefore remain in the image for the motion analysis method to identify them as blinking. A user seated before a monitor with a camera mounted on it may zoom the camera in or out so that the face comprises anywhere from roughly 15% to 100% of the image. For users with disabilities, the amount of zoom must take into account the degree to which the user may involuntarily move his or her head during the session.

The use of the simple correlation coefficient for tracking and degree of closure measurement has proven to be effective for this system. However, there are clear restrictions it imposes. For example, should the template selected be considerably larger than the actual eye in the image, then the eye comprises a smaller percentage of the template used for determining the degree of openness, and thus large movements of the eyelid have less impact than desired. Likewise, should a user begin to squint for an extended period of time, his or her open eye template becomes out of date, and the system may give faulty outputs until the tracker is lost for some reason and re-initializes itself. For this reason, the complementary motion analysis is valuable for reinforcing or discarding classifications made by the correlation component of the system.

6 Conclusions and Future Work

The algorithm and software presented in this paper constitute an alternative communication method that is suitable for people with severe disabilities. Results

ISBN 0-7695-1272-0/01 $10.00 (C) 2001 IEEE

demonstrate the Blink Link's ability to accurately distinguish between voluntary and involuntary blinks, an important consideration for a system controlled by facial gestures or cues. The system runs consistently at a frame rate of 27 to 29 fps, which is believed to be closer to the real-time goal of 30 fps than other blink detection systems previously designed. Prior knowledge of face location or skin color is not required, nor is any special lighting. The use of blink patterns as communication signals proposed in this work offers a novel approach to written word communication applications.

Some trackers used in human-computer interfaces for people with disabilities require the user to wear special transmitters, sensors, or markers. Such systems have the disadvantage of potentially being perceived as a conspicuous advertisement of the individual's disability. Since the Blink Link uses only a camera placed on the computer monitor, it is completely nonintrusive. The absence of any accessories on the user make the system easier to configure and therefore more user-friendly in a clinical or academic environment, as discussed in [15]. It is accommodating to most natural human movement because of its fast tracking and the automatic re-initialization feature.

Further testing of the system with users with disabilities is necessary to learn more about what is most comfortable and effective as an interface. Ideas for extending this project in the future include the development of the Huffman code blink system and a study of its feasibility. The Blink Link may lend itself very well to some combination with other assistive technologies to improve the bit-rate of communication for people with disabilities. It could also be used to augment interfaces based on natural language processing or American Sign Language recognition.

References

[1] L.-P. Bala, K. Talmi, and J. Liu. Automatic detection and tracking of faces and facial features in video sequences. In *Picture Coding Symposium*, Berlin, Germany, 1997.

[2] M. Betke, W. J. Mullally, and J. Magee. Active detection of eye scleras in real time. In *IEEE Workshop on Human Modeling, Analysis and Synthesis*, Hilton Head Island, SC, 2000.

[3] J. L. Crowley and F. Berard. Multi-modal tracking of faces for video communications. In *CVPR*, pages 640–645, 1997.

[4] J. Davis and A. Bobick. The representation and recognition of action using temporal templates. In *CVPR*, pages 928–934, 1997.

[5] The EagleEyes Project at Boston College. http://www.cs.bc.edu/˜eagleeye.

[6] J. Gips, M. Betke, and P. Fleming. The Camera Mouse: Preliminary investigation of automated visual tracking for computer access. In *RESNA 2000*, pages 98–100, 2000.

[7] J. Gips, P. DiMattia, F. X. Curran, and P. Olivieri. Using EagleEyes – an electrodes based device for controlling the computer with your eyes – to help people with special needs. In J. Klaus, E. Auff, W. Kremser, and W. Zagler, editors, *Interdisciplinary Aspects on Computers Helping People with Special Needs*. R. Oldenbourg, Vienna, 1996.

[8] http://www.ai.mit.edu/˜kgrauman.

[9] A. Haro, M. Flickner, and I. Essa. Detecting and tracking eyes by using their physiological properties, dynamics, and appearance. In *CVPR*, 2000.

[10] R. J. K. Jacob. What you look at is what you get. *Computer*, pages 65–66, July 1993.

[11] R. Kothari and J. Mitchell. Detection of eye locations in unconstrained visual images. In *IEEE Int. Conf. Image Processing*, pages 519–522, 1996.

[12] C. H. Morimoto and M. Flickner. Real-time multiple face detection using active illumination. In *IEEE Face & Gesture Recognition*, pages 8–13, 2000.

[13] T. Nakano, K. Sugiyama, M. Mizuno, and S. Yamamoto. Blink measurement by image processing and application to warning of driver's drowsiness in automobiles. In *IEEE Intelligent Vehicles*, pages 285–290, 1998.

[14] K. Ogawa and T. Okumura. Development of drowsiness detection system. In *IEEE Intelligent Transport Systems*, 1998.

[15] R. B. Reilly. Applications of face and gesture recognition for human-computer interaction. In *ACM Multimedia Conf. on Face/Gesture Recognition and their Applications*, pages 20–27, 1998.

[16] S. Singh and N. Papanikolopoulos. Vision-based detection of driver fatigue. http://www-users.cs.umn.edu/˜sasingh/research, May 2001.

[17] L. Stringa. Eyes detection for face recognition. *Applied Artificial Intelligence*, 7:365–382, 1993.

[18] Y. Tian, T. Kanade, and J. Cohn. Dual-state parametric eye tracking. In *IEEE Face & Gesture Recognition*, pages 110–115, 2000.

[19] S. Zhai, C. Morimoto, and S. Ihde. Manual and gaze input cascaded (MAGIC) pointing. In *CHI'99: ACM Conference on Human Factors in Computing Systems*, pages 246–253, 1999.

ISBN 0-7695-1272-0/01 $10.00 (C) 2001 IEEE