Univ Access Inf Soc (2003) 2: 1–15 / Digital Object Identifier (DOI) 10.1007/s10209-003-0062-x

Communication via eye blinks and eyebrow raises: video-based human-computer interfaces

K. Grauman[1], M. Betke[2], J. Lombardi[2], J. Gips[3], G.R. Bradski[4]

[1] Vision Interface Group, AI Laboratory, Massachusetts, Institute of Technology, 77 Massachusetts Avenue, Cambridge, MA, 02139, USA; E-mail: kgrauman@ai.mit.edu, http://www.ai.mit.edu/ ∼kgrauman
[2] Computer Science Department, Boston University, 111 Cummington St., Boston, MA 02215, USA; E-mail: {betke, questor}@cs.bu.edu, http://www.cs.bu.edu/faculty/betke
[3] EagleEyes, Computer Science Department, Boston College, Fulton Hall, Chestnut Hill, MA 02467, USA; E-mail: gips@bc.edu, http://www.cs.bc.edu/ ∼gips
[4] Vision, Graphics and Pattern Recognition, Microcomputer Research Laboratory, Intel Corporation, SC12-303, 2200 Mission College Blvd., Santa Clara, CA 95054-1537; E-mail: gary.bradski@intel.com

Published online: [redacted] 2003 –

Abstract. Two video-based human-computer interaction tools are introduced that can activate a binary switch and issue a selection command. "BlinkLink," as the first tool is called, automatically detects a user's eye blinks and accurately measures their durations. 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 second tool, "EyebrowClicker," automatically detects when a user raises his or her eyebrows and then triggers a mouse click. Both systems can initialize themselves, track the eyes at frame rate, and recover in the event of errors. No special lighting is required. The systems have been tested with interactive games and a spelling program. Results demonstrate overall detection accuracy of 95.6% for BlinkLink and 89.0% for EyebrowClicker.

Keywords: Computer vision – Assistive technology – Camera-computer interface

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 traumatic brain injuries and cerebral palsy to multiple sclerosis (MS) or amyotropic lateral sclerosis (ALS). The access to information and enhanced communication that assistive technology provides is both practical and empowering for individuals with disabilities. A secondary goal of our work is to provide new tools to access computing devices for the general population that lead to a natural and enjoyable interaction with the computer.

We propose robust, accurate algorithms to detect eyes and eyebrows, measure the duration of blinks and eyebrow raises, and interpret them in real time to control a nonintrusive human-computer interface. BlinkLink uses methods that employ 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. EyebrowClicker uses similar image analysis methods to detect the changing appearance of the eyebrows and measure the distance between eyebrows and eyes. Both systems are designed to initialize themselves automatically and adjust for changes in the user's position in *depth*, i.e., the user's distance to the video camera and computer screen. Both systems are user independent, i.e., they provide a general scheme for interpreting *any* user's blink or eyebrow motion.

With EyebrowClicker, a user who is capable of raising his or her eyebrows can generate mouse clicks or

MS ID: UAIS0062
23 May 2003 14:10 CET

other selection commands to operate software applications requiring such input. EyebrowClicker can thus augment the traditional keyboard and mouse interfaces. BlinkLink is designed for people with severe disabilities who are nonverbal and have very limited muscle control but are capable of blinking voluntarily. Such users can generate selection commands through their eye blinks to operate switch-based scanning programs and on-screen keyboards [49].

The two systems use various image processing and computer vision techniques in combination. Eye blink motion is used in both systems 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. In BlinkLink, 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 video frame. In EyebrowClicker, eye and eyebrow templates are used for correlation-based tracking and their distances relative to the size of the features are measured.

An array of techniques have been explored for locating eyes and eyebrows in images and eye blink detection. Methods for detecting the eyes include the use of gradient flow fields [37], color-based techniques for detection of the eye sclera [5], horizontal gradient maps of a skin-colored region [48, 51], and pupil detection using infrared or other special lighting [2, 31, 40, 54]. References [1, 7, 13, 17, ?, 21, 38, 40, 50] explain various face and head tracking techniques previously employed. Temporal differencing is often used to segment moving regions of interest from a stable background [15, 16]. Methods for analyzing the eye and its closure motion are suggested in References. [1, 15, 35, 41, 42, 48, 52]. A blink detector has been developed to detect drowsy drivers [41]. Facial-feature trackers [4, 12, 24, 45] have been designed to provide video-based interfaces for people with disabilities. Eyebrow tracking is often used for determining the facial expression of the user [8, 22, 29]. We are not aware of any papers that address the issues of video-based communication interfaces that operate on eye blinks or eyebrow raises. 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 own preliminary work is described in [30] and [39].

Our contribution is to provide two real-time systems that consistently run at frame rate, are completely nonintrusive, and require no manual initialization, prior face detection, or special lighting. The BlinkLink 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. BlinkLink has also been tested for recognizing substantial deformations of other features, for example the mouth. EyebrowClicker can reliably interpret the motion of the eyebrows and detect a raised eyebrow based on its distance to the eye and duration of the motion.

Alternative communication systems for computer users with severe disabilities are switch-based scanning interfaces (e.g., [44]). Activating the switch with the tongue or hand initiates a scan through a matrix of icons, letters, words, or phrases. The scan proceeds automatically, pausing at each matrix entry for a specific time interval. A matrix entry can be selected by hitting the switch.

People with severe disabilities who have retained the ability to move their heads voluntarily can use head-mounted or camera-based tracking devices (e.g., [11, 23, 26, 45]) to enter text and other data into a computer. Other popular systems are gaze estimators based on measuring the electro-oculographic potential or corneal reflections [20, 32, 46, 52–54]. The goal of introducing eye blink detection functionality in a camera-based system is to provide another point of computer access for those users who may not be capable of motor controls that some of the above methods demand and therefore must rely on a human interpreter. Jean-Dominique Bauby, for example, had no method of communication except for blinking after he suffered a severe stroke in the brain stem. He wrote about his condition in his memoir *The Diving Bell and the Butterfly* [3]. He composed each passage mentally and then dictated it, letter by letter, to a caregiver. The caregiver recited a frequency-ordered alphabet until Bauby selected a letter by blinking.

The two video-based interaction systems were tested by many computer users in various experiments. We report a system accuracy of 95.6% for our experiments with BlinkLink and 89.0% with EyebrowClicker that included subjects who could blink voluntarily and who comprehended the tasks without difficulty. User testing for BlinkLink was performed at Boston College's Campus School for children with various severe disabilities. Currently, children there rely on two systems as mouse replacements: the CameraMouse system uses a video camera to perform facial-feature tracking [4, 12, 24–27], and the EagleEyes system measures the user's electro-oculographic potential to estimate gaze direction [20, 28]. Children use the systems to spell out messages, play games, and even participate in distance learning programs on the Web. Increasingly these earlier systems are being installed at off-site locations in homes and schools.

In feature-tracking systems such as CameraMouse [4], some small section of the face is tracked and used to generate corresponding mouse motion. A user makes a selection or issues a mouse click by dwelling

MS ID: UAIS0062
23 May 2003 14:10 CET

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 [33, 34]. One of the selection methods proposed in this work, EyebrowClicker or BlinkLink, may be used in conjunction with such feature-tracking methods to provide a more active means of making selections. A prolonged blink or eyebrow raise is a more emphatic way of indicating voluntary selections.

This paper is organized as follows. The image analysis methods employed in the two interface systems are described in the main part of the paper. The hardware specifications of the interfaces and a description and discussion of the experiments follow. The paper concludes with a summary and discussion of future work.

2 Methods

In this section, an overview of the BlinkLink interface is given, followed by a description of the video analysis methods employed. Then an overview of EyebrowClicker and a description of its methods are presented.

2.1 BlinkLink*: 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 video 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

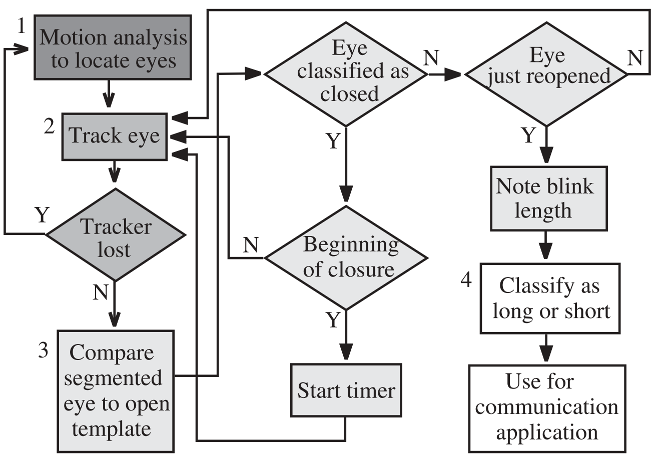

Fig. 1. Components of BlinkLink: 1. Motion analysis (shown in *dark gray*), 2. Eye tracking (*gray*), 3. Blink detection and duration analysis (*light gray*), and 4. Interpretation (*white*)

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 reinitialized by repeating motion analysis on the subsequent involuntary blinks.

2.2 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 [36]. The difference image $D(x, y, t) = |I(x, y, t) - I(x, y, t-1)|$ is formed from the previous frame image $I(x, y, t-1)$ and the current frame image $I(x, y, t)$ for all pixels (x, y) 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 [47] 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).

Finally, candidate eye "blobs" are extracted by labeling the connected components in the preprocessed 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 pixel 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 nonblink 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 [19] between

MS ID: UAIS0062
23 May 2003 14:10 CET

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

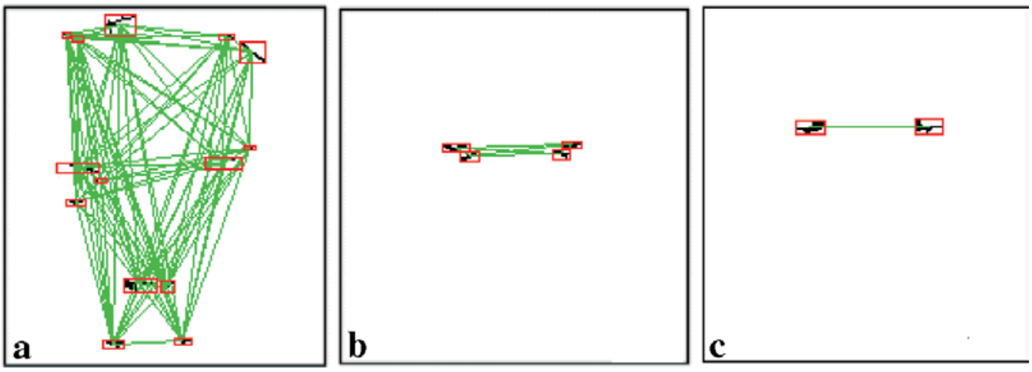

Fig. 3. Thresholded, segmented difference image showing arbitrary motion (**a**), two candidate pairs falling below the Mahalanobis distance threshold (**b**), and one candidate pair identified as a blink (**c**). *Boxes* bound regions of motion, and *green lines* connecting the boxes indicate component pairings

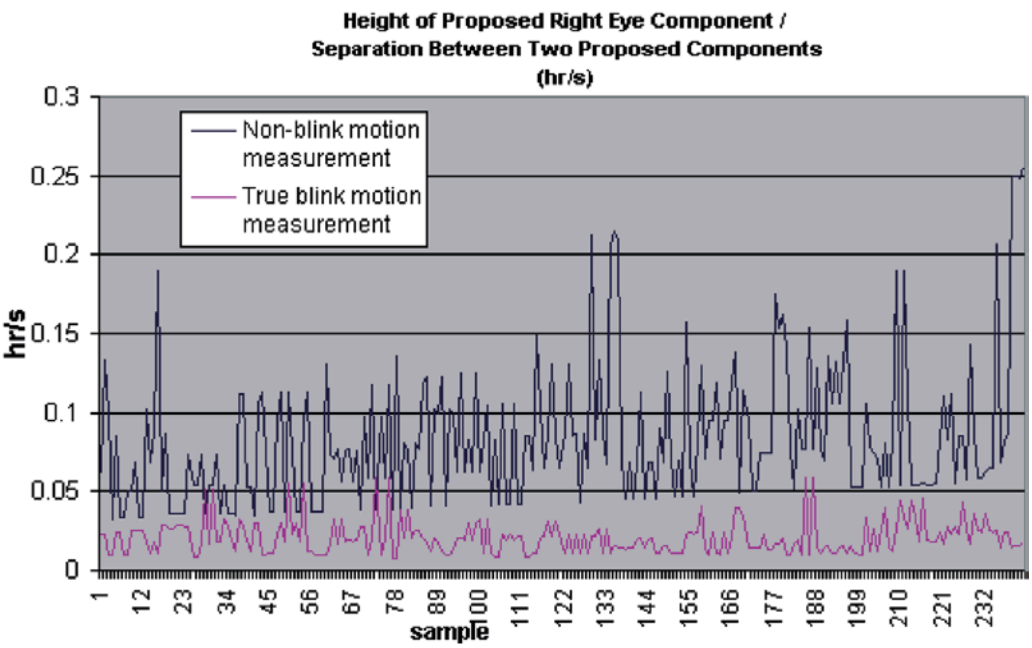

Fig. 4. Example of a filter applied to candidate eye component pairs. Values associated with instances of true eye blinks are significantly lower than those of nonblink motion in the scene. A filter can therefore be used to quickly discard candidate motion blobs that do not describe eye blinks

MS ID: UAIS0062
23 May 2003 14:10 CET

the candidate pair's vector of properties $\mathbf{p}$ and the mean vector of blink-pair properties μ, where

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

The mean vector μ and covariance matrix Σ for computing distance d are produced by manually identified blink-pairs at different depths and face orientations during training. The accumulation of these representative true blink-pairs is an offline training process that need be done only once before the system is installed. We collected the blink-pairs by having about 30 users sit perfectly still in front of the camera several times, each time at different depths and with their faces at different in-plane rotations, and blink repeatedly. In this way we could ensure that all motion detected in the frames would be from blinks and thus would produce true blink-motion examples to be used as a model.

The Mahalanobis distance measure was chosen because it provides an optimality criterion based on the least-squares error. It computes the sum of squared distances between the measured properties of the motion components and their corresponding sample means, weighted by the respective sample covariances. The weighting addresses the potential correlation between the features. The measured features are classified as anthropomorphically infeasible if the error is larger than a threshold. Motion-pairs having errors less than the threshold are classified as blink-pair candidates. For a given frame, if there exists only one pair of motion components with anthropomorphically feasible features, 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 the pair with the smallest error is considered. The steps of the motion analysis phase are summarized in Fig. 5.

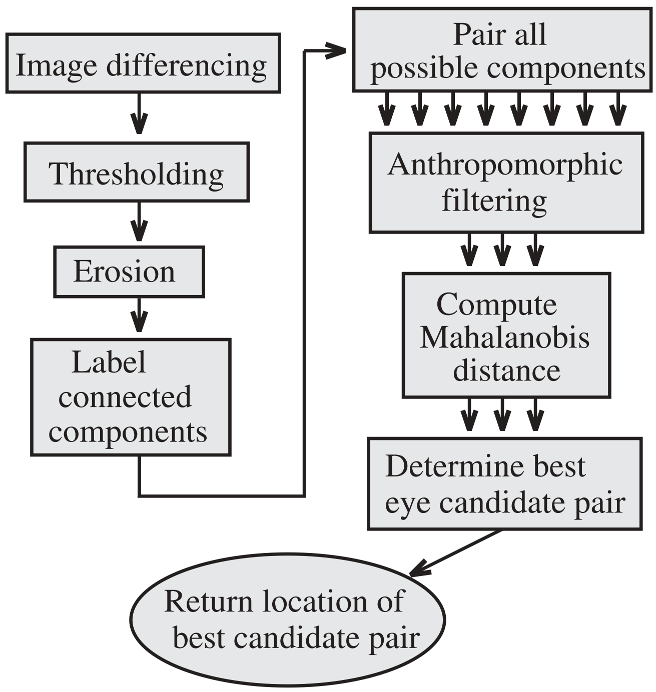

Fig. 5. Details of motion analysis phase

Relying on the assumption that the motion of the best candidate pair was caused by the quick closing and opening of the eyes, a template of the open eye is 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 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.

2.3 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 up-

MS ID: UAIS0062
23 May 2003 14:10 CET

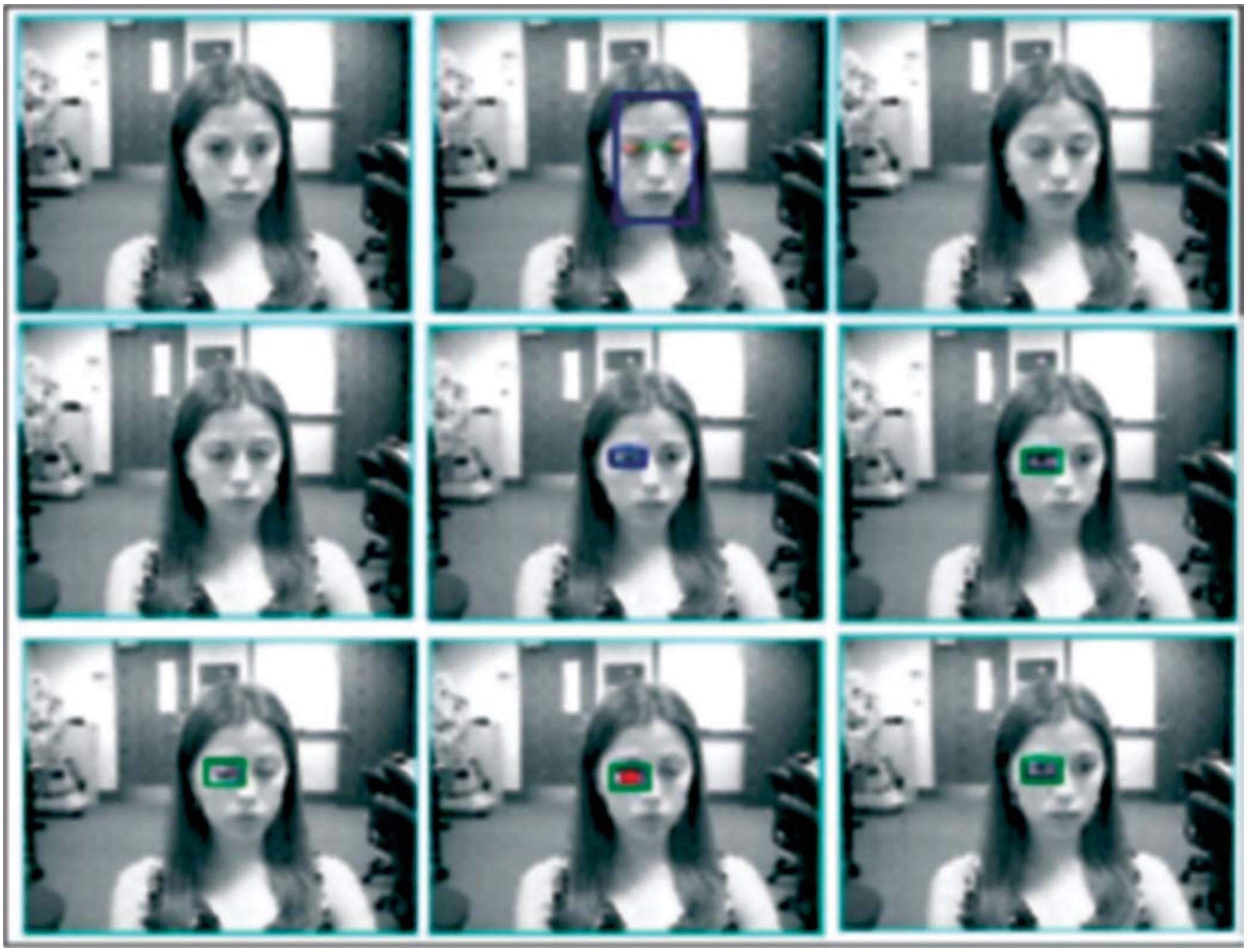

Fig. 6. Intermittent frames from a sequence during the motion analysis phase when the template is being found automatically by the user's first several natural blinks. *Rectangles* around the face indicate that blink-like motion was detected. The *small rectangle* that appears around the eye three frames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

dated nearly 30 times per second and remains accurate barring dramatic, sudden head movements or significant changes in depth. For these events, it is critical that the tracker declare itself lost and reinitialize 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. 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 shows a sequence of frames during the motion analysis phase in which the subjects' eyes are detected and tracked.

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

Experiments comparing the correlation scores of the actual eye and its closed template with the scores of the actual eye and its open template confirmed that both methods succeed in identifying blinking. 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.

MS ID: UAIS0062
23 May 2003 14:10 CET

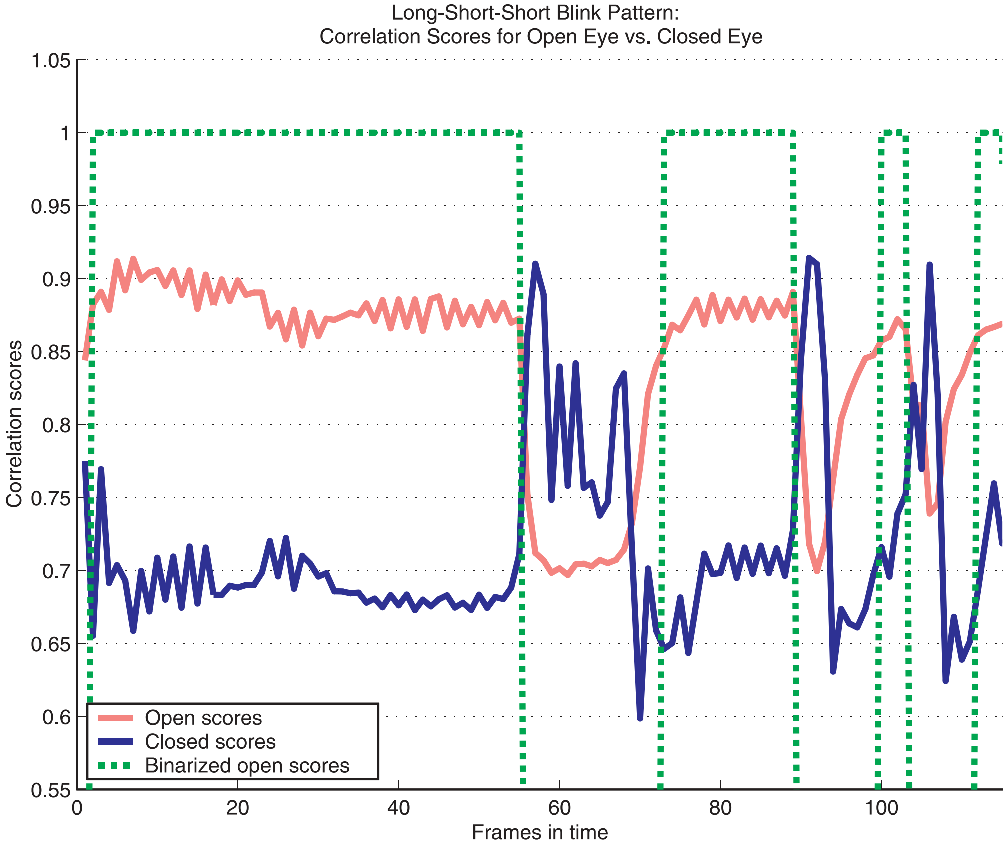

Fig. 7. Correlation scores over time comparing the user's eye at each frame to both the open-eye template and the closed-eye template. The open-eye scores present a waveform indicating the captured blink pattern: *long, short, short.* Such samples were collected and used to identify an effective threshold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

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 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 *not found* 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 obtained for system training 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. Given the observations from the training data, threshold values $O = 0.85$ and $F = 0.55$ were chosen for the BlinkLink interface.

2.5 EyebrowClicker*: system overview*

EyebrowClicker initializes itself using the blinks of a user to determine the locations of the eyes and eyebrows. Once these locations are detected, each feature is tracked individually. If the user raises his or her eyebrows, the program notes that the distance between the eyes and eyebrows increases and sends the selection command, a mouse click, to the computer (see Fig. 8).

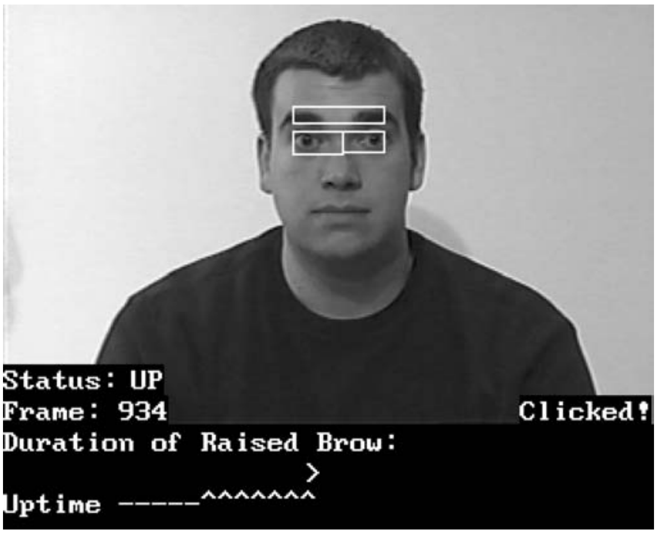

Fig. 8. A user of EyebrowClicker immediately after issuing a "click" as a selection command. The *bar* on the bottom displays the duration of the current eyebrow-raising action

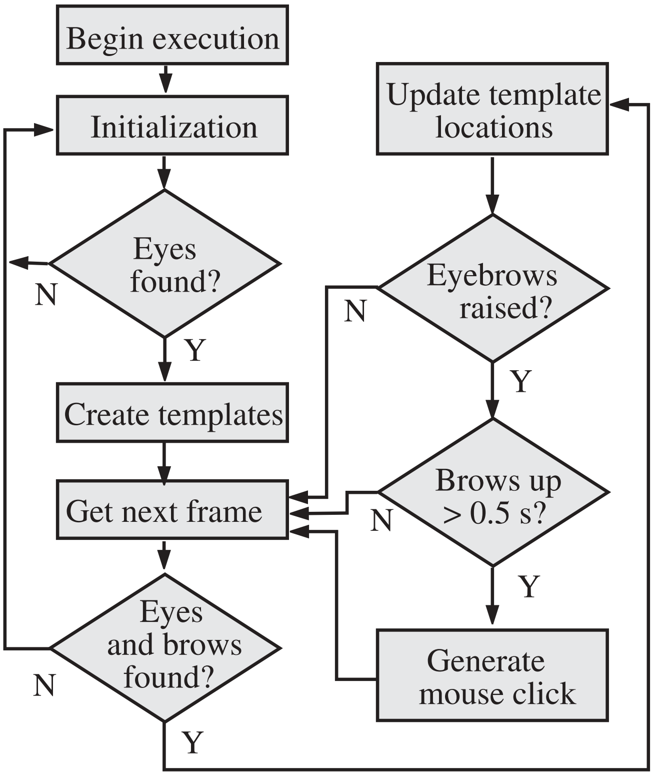

Fig. 9. Components of EyebrowClicker

EyebrowClicker has two phases: the initialization/recovery phase and the tracking phase. A flowchart of the system is given in Fig. 9.

2.6 EyebrowClicker*: initialization and recovery phase*

Like BlinkLink, EyebrowClicker looks for a user's first several natural blinks to locate the user's eyes. To detect the blinks, difference images of the current and previous frames are created, $D(x,y,t) = I(x,y,t) - I(x,y,t-1)$, and the motion energy image (MEI) is computed [10]. The MEI $E_\tau(x,y,t)$ is defined as the union of a sequence of τ binary difference images:

$$E_\tau(x,y,t) = \bigcup_{i=0}^{\tau-1} D(x,y,t-i). \tag{2}$$

A sequence length of $\tau = 60$, which corresponds to 2 s of video, is used. In the MEI, two large regions of "motion energy" occur where the blinking took place. It may also contain scattered noise caused by other motion in the scene. The MEI is further processed as described for BlinkLink in Sect. 2.2 to estimate the locations and size of the user's eyes in the scene. Based on these estimates, grayscale templates for the eyebrow pair and each of the eyes are cropped out of the current image frame and the tracking phase is entered.

The above methods are also used in the recovery phase in case tracking fails at some point in time and the system needs to reinitialize itself.

2.7 EyebrowClicker*: tracking phase*

After the templates for the eyebrow pair and each of the eyes are obtained, EyebrowClicker enters its tracking phase. The last known location of each of the templates is used to begin the search in the current frame. The system searches the local area using the normalized correlation coefficient to determine the best match (see Fig. 10). If the best match is below the 0.5 threshold, the tracker is assumed to be lost and the system reinitializes to obtain better templates.

If all three templates are successfully captured, the program uses anthropomorphic information about the face to further protect against any incorrect tracking. If the relative locations of the templates are anthropomorphically infeasible, e.g., in the extreme case one eye template is above the eyebrow template, the tracking phase ends and reinitialization occurs. Significantly overlapped or spaced templates will also force the program to recover itself.

When the tracker determines that the eyes and eyebrows are in a state consistent with human facial structure, it computes the ratio

$$T_{\text{current}} = \frac{(y_{\text{left eye}} + y_{\text{right eye}})/2 - y_{\text{eyebrows}})}{(h_{\text{left eye}} + h_{\text{right eye}})/2} \tag{3}$$

where $y_{\text{eyebrows}}, y_{\text{left eye}}$, and $y_{\text{right eye}}$ represent the y-coordinates of the respective templates and $h_{\text{left eye}}$ and $h_{\text{right eye}}$ the height of the respective templates. If T_{current} is larger than some threshold T, the system determines that the eyebrows are raised. A threshold of $T = 1.25$ is sufficient to remove most jitter caused by eye movement yet still small enough so that all users are able to surpass this bound easily.

Once the threshold T has been crossed, a timer is set, and if the user's eyebrows remain up for a constant number of milliseconds – we used 500 ms – the selection command is delivered to the system (see Fig. 11). This additional timing threshold prevents occasional jitter from issuing a false click.

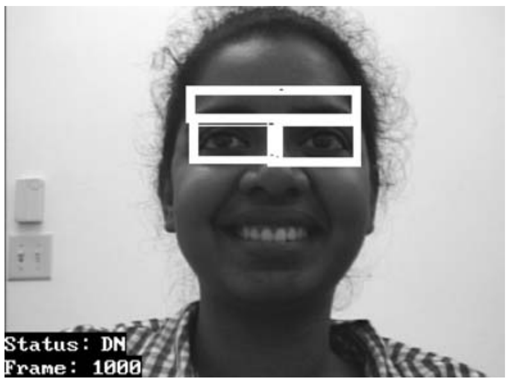

Fig. 10. The white pixels denote the localized areas where the search proceeds for each of the templates

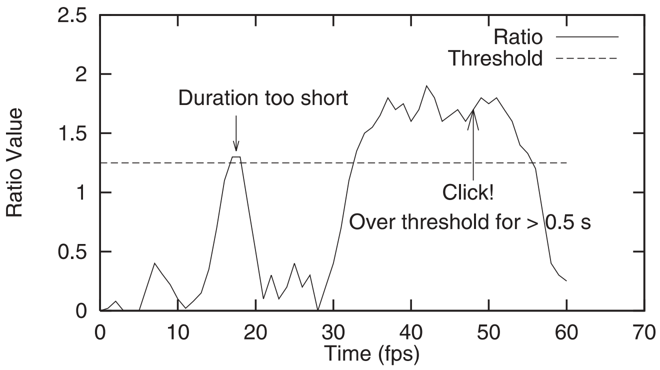

Fig. 11. Values of Eq. (3) are plotted while a user raises and lowers her eyebrows. The region where the graph crosses the threshold indicates jitter caused by a user blink

Fig. 12. BlinkLink in use: the video camera is placed on top of the left monitor, which displays the application software. The right computer monitor (not required) is used to check the status of the system. A user plays an arcade game based on scanning software where the objective is to throw a baseball at plates on a shelf. A long blink causes a ball to be thrown. The user must time the blink so that it coincides with the highlighting of a plate

After a click has been issued for the current eyebrow motion, the system prevents any additional clicking actions until the user's eyebrows return to a relaxed state. This avoids a situation in which one eyebrow movement can result in a potentially infinite number of clicks in rapid succession (see Fig. 11).

3 Materials

For development and testing of the interface systems, a Sony EVI-D30 color video CCD camera was used with a Matrox Meteor II image capture board. Grayscale images are processed at 320×240 pixels. BlinkLink was implemented on a Windows NT Workstation with two 1-GHz processors with 256 MB RAM, EyebrowClicker with a single 866-MHz processor with 1-GB RAM. The systems were implemented with Open CV, Intel's Open Source Computer Vision Library [43] that interfaces the Matrox frame grabber with Intel's Image Processing Library [43].

Testing of BlinkLink at the Boston College Campus School with people with disabilities was done on an 800-MHz single-processor PC with 128 MB RAM. When possible, two displays are helpful but not necessary when running BlinkLink, as this allows one to monitor the status of the eye tracker. Figure 12 shows BlinkLink in use.

During the experiments with EyebrowClicker, the camera was placed on the table directly in front of the monitor. Each of the users was seated approximately 2–3 feet from the front of the monitor (see Fig. 13). Regular overhead fluorescent lighting was used.

4 BlinkLink: experiments and discussion

The BlinkLink system has been tested for its accuracy as well as its usability as an input device. Test subjects need a certain level of physical and mental facility to use the system. For example, the children at Boston College with severe cerebral palsy cannot blink voluntarily.

Fig. 13. A user accessing the computer with EyebrowClicker. The camera is placed in front of the monitor but below the screen

A teenager with a traumatic brain injury was not able to understand the tasks we asked him to perform.

The results, summarized in Table 1, are based on sessions with 15 different subjects who could blink voluntarily and comprehended the tasks without difficulty. Several videos of our experiments are provided on the Web [9] and include a subject with glasses, two people in the field of view of the camera competing for the tracker, a subject performing quick head motions, and distractions by motion in the background.

In order to measure the accuracy of eye blink detection, video sequences were captured of each user sitting between 2 and 4 ft from the camera. The users were asked to blink naturally but frequently and exhibit mild head movement. Each sequence was processed by the BlinkLink system in real time. Processed images were saved

MS ID: UAIS0062
23 May 2003 14:10 CET

Table 1. Summary of results

Overall accuracy of blink detection	95.6%
System sensitivity of blink detection	98.0%
Long/short classification accuracy	93.0%
Usability score as an input device	93.6%
Average frame rate	28 fps

and manually examined offline to determine precisely how the system had interpreted the data.

The system sensitivity of blink detection was 98%: out of the 204 actual blinks in the sequences 200 were detected and only 4 missed. False positives were encountered five times, yielding an overall detection accuracy of 95.6%.

Beyond simple detection accuracy, it was important to test BlinkLink'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 BlinkLink system correctly detected all but two of the combined long and short blinks, yielding a 98% rate of accuracy for detection for these samples. Ninety-three percent 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.

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 mouse location. While the eye tracking information may be used to generate mouse movement, for BlinkLink 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 BlinkLink. 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 a selecting click 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 the game Frog 'N Fly, 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 (see top of Fig. 14). In another game, images of familiar objects are shown, and the user must blink when the image of a matching object is shown (see bottom of Fig. 14).

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

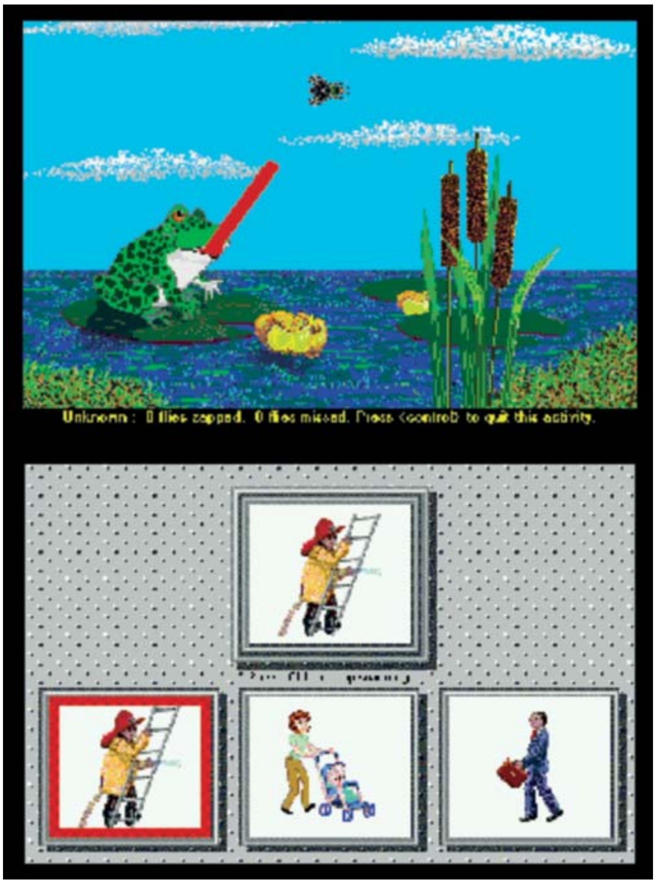

Fig. 14. Sample games testing reaction time (*top*) and visual matching abilities (*bottom*). The user must blink voluntarily when a fly appears in order to have the frog catch it with its tongue (*top*). The red outlining box cycles through the options, and the user blinks when the matching image is outlined (*bottom*)

long enough to meet the voluntary length threshold, or false positives 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 s (see Fig. 15). We measured a communication rate of about 9 s per letter for our novice users. The speed is greatly determined by the speed of the scanning interface, i.e., how quickly the software cycles through the grid of options. This parameter may be adjusted for more experienced users.

The system is equipped to handle head movement, rotation in the image plane, and as much horizontal head turning or vertical nodding as is necessary such that neither eye is completely occluded. Should the tracker become lost because of a sudden acceleration of the head, it is reinitialized 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, if the template selected is 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 reinitializes itself. For this reason, the complementary motion analysis is valuable for reinforcing or discarding classifications made by the correlation component of the system.

In both the blink and eyebrow-raise systems, one component of detection relies on motion information calculated from changes in brightness. Thus, while special lighting or special cameras are not required, we do require a fairly *constant* imaging setting. Large, sudden changes from moving the camera or a light source mid-use will influence the system. For best performance, the system should be used in the same imaging setting in which it was trained. In our experiments in both a classroom with natural light from large windows as well as a laboratory with fluorescent lighting, lighting consistency was adequate. There is also an inherent sensitivity to specularities, as is the standard case with most video-based systems that utilize brightness or color information in the image. A specularity occurs where the light source rays intersect with certain surfaces, such as a user's eyeglasses, and reflect in such a way as to cause a bright spot in the image. Such bright spots hide the true texture or color information and thus may mislead our method. The system was able to detect blinks from the user with glasses, although in such cases the tracker will be more sensitive to head rotations since they may induce new specularities. Thus, it will necessarily reinitialize itself more frequently.

When the eye tracker is lost, it reinitializes when the user next blinks, which is typically within a few seconds. For an application that cycles indefinitely, for example the spelling program shown in Fig. 15, this merely slows the communication rate but does not introduce errors. However, for other application programs, such as games testing reaction time, the loss of the tracker affects the user's performance if the application program cannot be halted while our system is being reinitialized.

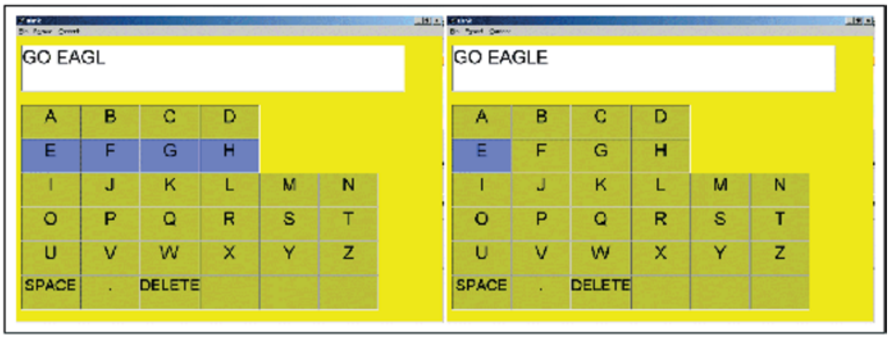

Fig. 15. The scanning spelling program highlights each consecutive row of letters. A blink causes it to highlight each consecutive letter in that row, and another blink causes the letter to be printed on top of the screen. Here, the second row is highlighted in the left image. A blink selects the letters E–H to be scanned from left to right; another blink selects the letter E in the right image

5 EyebrowClicker: experiments and discussion

EyebrowClicker was tested for its accuracy and usability as an input device by six different subjects who had control of the appropriate facial muscles and comprehended the tasks.

A video demonstrating our experiments is provided on the Web [6]. The users practiced with the system before testing. Each user began by watching himself raise and lower his eyebrows and determining how to use EyebrowClicker. During this training, the users simply executed mouse clicks whenever they desired, thereby adjusting to the system. We also asked the users to experiment with the amount of yaw, pitch, and roll rotations they could perform with their head and still have EyebrowClicker function as expected.

After the user felt comfortable using EyebrowClicker, usually within 2–3 min or less of training, we started the actual test. For the test, we used the Frog 'N Fly game as shown in Fig. 16. With EyebrowClicker as the human-computer interface, the user must raise his eyebrows to cause the frog to catch the fly as soon as a circle appears around the fly. Each user was asked to catch the flies that appeared during 3 min of the game.

The six subjects caught 243 out of 273 flies, which represents an 89.0% success rate. Two reinitializations occurred during the middle of the game. The reinitialization caused the number of misses to be inflated, because Frog 'N Fly could not be paused during reinitialization. When the tracker became lost, the user missed two or even three flies during the time the tracker needed to recover. Some of the misses were due to the system reinitializing itself. Discounting these misses, the success rate becomes slightly better. Only two clicks were falsely generated throughout all trials. We therefore achieved one of our primary goals in creating the system – an extremely low false-positive rate.

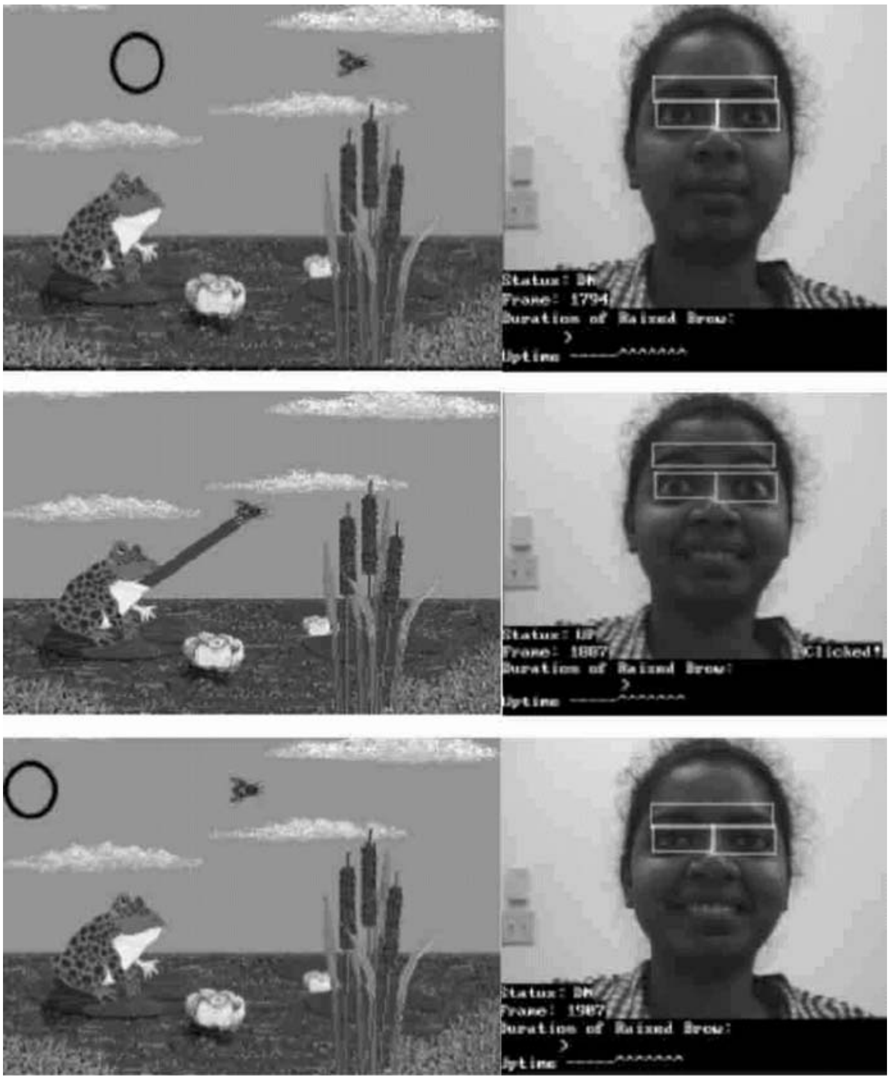

Fig. 16. Screenshots of the Frog 'N Fly game and a subject playing the game using EyebrowClicker

MS ID: UAIS0062
23 May 2003 14:10 CET

Users gain speed as they get accustomed to the system and accumulate practice with the application. For example, when the users first tried the system, the initialization phase lasted up to a couple of minutes. After the users played with the frog game for several minutes and purposely went through the process of reinitialization a few times, they were able to force the system to initialize within a few seconds.

EyebrowClicker functions at 29.5 frames per second during all segments of execution on our test machine. It never used more than 21% of the CPU resources, demonstrating the potential for concurrent use with other programs without tremendous burden on the machine.

EyebrowClicker places restrictions on the freedom of movement of the user. Our results indicate that the algorithm does not tolerate shaking and turning of the user's head, i.e., rotations measured by the yaw and pitch angles. However, bending the head forward or raising it, measured by the roll angle, is tolerated up to approximately 20° in most users, as long as the eyebrows do not become occluded due to head wear or hair. Head translation parallel to the image plane is allowed with complete freedom. In addition, the system will detect when the user moves toward the camera or away from it. The system then reinitializes itself to obtain new templates. This is desirable since the size of the face changes in the image and the templates must be updated appropriately.

If the user is in the middle of a blink when the system obtains copies of the templates, EyebrowClicker's eye tracking module does not "lock on" to the eyes as well. The eye tracker will still stay close to the precise location of the eye but will move around the area slightly. This does not generate false clicks, except in the unlikely case that a user closes his eyes and keeps them shut.

6 Comparison of BlinkLink and EyebrowClicker

EyebrowClicker is preferred over BlinkLink by some of our subjects who were able to use both systems because it allows them to look at the screen while issuing a selection command. They perceived the moment of eye closure, required by BlinkLink, as disruptive. Other users, however, preferred BlinkLink because they considered a sequence of blinking operations to be less tiring than a sequence of eyebrow raises.

7 Discussion of communication strategies using visual cues

The experiments described above used BlinkLink and EyebrowClicker with application software based on a switch-based scanning strategy. Other communication strategies using eyebrow raises or blinks as input are possible.

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 [14] 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. Future user testing is necessary to determine the practicality of the approach.

Facial features other than eyes or eyebrows may also be used to generate a switch. 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. In preliminary tests, the subjects 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. Our subjects' response shows that for some people the motion of the mouth is easier to control than the eyes or eyebrows and is thus a better input method. Given the broad range of differing abilities possessed by users of assistive technology, multiple options for usage are certainly valuable.

8 Conclusions and future work

The video-based interfaces presented in this paper constitute alternative communication methods that can replace the mouse in applications that solely require selection commands. Results demonstrate BlinkLink's ability to accurately distinguish between voluntary and involuntary blinks. Experiments with EyebrowClicker show that eyes and eyebrows can be detected and tracked automatically and eyebrow raises can be recognized with high accuracy. Prior knowledge of face location or skin color is not required, nor is any special lighting. Both systems run consistently in real time, an important consideration for systems controlled by facial gestures or cues. The limited need for computational resources makes both interfaces viable for concurrent use with other application programs on modern computers.

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 presented interfaces use only a camera placed on or near

MS ID: UAIS0062
23 May 2003 14:10 CET

the computer monitor, they are completely nonintrusive. The absence of any accessories on the user makes the systems easier to configure and therefore more user-friendly in a clinical or academic environment, as discussed in [45]. They are accommodating to most natural human movement because of their fast tracking and the automatic self-correcting capabilities.

Ideas for extending this project in the future include the development of the Huffman code blink system and a study of its feasibility. The offline training phase could be extended with an online calibration phase that learns the specific user's facial properties or preferred blink patterns. Both BlinkLink and EyebrowClicker may lend themselves very well to some combination with other assistive technologies to improve the bit rate of communication for people with disabilities. They could also be used to augment natural-language interfaces to recognize both spoken and signed language. An eyebrow raise, for example, is an important grammatical tool in American Sign Language (ASL) to indicate a question. In future work, we will incorporate our eyebrow-raise detector in a more general interface for computerized translation of ASL.

Acknowledgements. Thanks to the subjects, caregivers, and graduate students who helped with the testing. Thanks to Hunter Larson of Boston College for programming the scanning spelling application. Figures 14 and 16 are courtesy of Simtech Publications. The work has been supported by the Office of Naval Research and the National Science Foundation, grants EIA-9871219, IIS-0093367, and EIA-0202067.

References

1. Bala L-P, Talmi K, Liu J (1997) Automatic detection and tracking of faces and facial features in video sequences. In: Abstracts of the picture coding symposium, September 1997, Berlin, Germany
2. Baluja S, Pomerleau D (1994) Non-intrusive gaze tracking using artificial neural networks. Technical report CMU-CS-94-102, Computer Science Department, Carnegie Mellon University, Pittsburgh
http://www.ri.cmu.edu/pubs/pub_2765.html
3. Bauby J-D (1997) The diving bell and the butterfly. Knopf, New York
4. Betke M, Gips J, Fleming P (March 2002) The Camera Mouse: visual tracking of body features to provide computer access for people with severe disabilities. IEEE Transactions on neural systems and rehabilitation engineering, 10(1):1–10
5. Betke M, Mullally WJ, Magee J (June 2000) Active detection of eye scleras in real time. In: Abstracts of the IEEE workshop on human modeling, analysis and synthesis, Hilton Head Island, SC, Technical report BCCS-99-04
6. Human-Computer Interfaces Web page at Boston University. http://www.cs.bu.edu/faculty/betke/research/hci.html
7. Birchfield S (2000) Elliptical head tracking using intensity gradients. In: Abstracts of the IEEE computer vision and pattern recognition conference, Hilton Head Island, SC, IEEE Computer Society, pp 232–237
8. Black MJ, Yacoob Y (June 1995) Tracking and recognizing rigid and non-rigid facial motions using local parametric models of image motions. In: Abstracts of the 5th international conference on computer vision, Cambridge, MA, pp 374–381
9. Blink Detection Videos (2001)
http://www.ai.mit.edu/ ∼kgrauman
10. Bobick A, Davis J (2001) The recognition of human movement using temporal templates. IEEE Transactions on pattern analysis and machine intelligence, 23(3):257–267
11. Chen YL, Tang FT, Chang WH, Wong MK, Shih YY, Kuo TS (1999) The new design of an infrared-controlled human-computer interface for the disabled. IEEE Transactions on rehabiliation engineering, 7(4):474–481
12. Cloud RL, Betke M, Gips J (2002) Experiments with a camera-based human-computer interface system. In: Abstracts of the 7th ERCIM workshop on user interfaces for all, Paris, pp 103–110
13. Comaniciu D, Ramesh V (2000) Robust detection and tracking of human faces with an active camera. In: Abstracts of the IEEE international workshop on visual surveillance, Dublin, pp 11–18
14. Cormen TH, Leiserson CE, Rivest RL (1990) Introduction to algorithms. MIT Press/McGraw-Hill, New York
15. Crowley JL, Berard F (1997) Multi-modal tracking of faces for video communications. In: Abstracts of the 1997 IEEE conference on computer vision and pattern recognition, Puerto Rico, June 1997, pp 640–645
16. Davis J, Bobick A (June 1997) The representation and recognition of action using temporal templates. In: Abstracts of IEEE conference on computer vision and pattern recognition, Puerto Rico, pp 928–934
17. De La Torre F, Yacoob Y, Davis L (2001) A probabilistic framework for rigid and non-rigid appearance based tracking and recognition. In: Abstracts of the 4th IEEE international conference on automatic face gesture recognition, Grenoble, France, pp 491–498
18. DeCarlo D, Metaxas D (June 1996) The integration of optical flow and deformable models with applications to human face shape and motion estimation. In: Abstracts of the 1996 IEEE Computer Society conference on computer vision and pattern recognition, San Francisco, IEEE Computer Society, pp 231–238
19. Duda RO, Hart RE, Stork DG (2001) Pattern classification, 2nd edn. Wiley, New York
20. The EagleEyes Project at Boston College.
http://www.cs.bc.edu/ ∼eagleeye
21. Edwards GJ, Taylor CJ, Cootes TF (1998) Learning to identify and track faces in image sequences. In: Abstracts of the international conference on face and gesture recognition, Nara, Japan, pp 260–265
22. Essa IA, Pentland A (1995) Facial expression recognition using a dynamic model and motion energy. In: Abstracts of the 5th international conference on computer vision, Cambridge, MA, pp 360–367
23. Evans DG, Drew R, Blenkhorn P (2000) Controlling mouse pointer position using an infrared head-operated joystick. IEEE Transactions on rehabilitation engineering, 8(1):107–117
24. Fagiani C, Betke M, Gips J (2002) Evaluation of tracking methods for human-computer interaction. In: Abstracts of the IEEE Workshop on applications in computer vision, Orlando, pp 121–126
25. Gips J, Betke M, DiMattia PA (2001) Early experiences using visual tracking for computer access by people with profound physical disabilities. In: Abstracts of the 1st international conference on universal access in human-computer interaction, New Orleans
26. Gips J, Betke M, Fleming P (July 2000) The camera mouse: preliminary investigation of automated visual tracking for computer access. In: Abstracts of the Rehabilitation Engineering and Assistive Technology Society of North America 2000 annual conference, Orlando, pp 98–100
27. Gips J, DiMattia P, Betke M (August 2002) Collaborative development of new access technology and communication software. In: Abstracts of the 10th biennial conference of the International Society for Augmentative and Alternative Communication (ISAAC 2002), Odense, Denmark
28. Gips J, DiMattia P, Curran FX, Olivieri P (1996) Using EagleEyes – an electrodes based device for controlling the computer with your eyes – to help people with special needs. In: Klaus J, Auff E, Kremser W, Zagler W (eds) Interdisciplinary aspects on computers helping people with special needs. R. Oldenbourg, Vienna

29. Gokturk SB, Bouguet J-Y, Tomasi C, Girod B (2002) Model-based face tracking for view-independent facial expression recognition. In: Abstracts of the 5th IEEE international conference on automatic face and gesture recognition, Washington, DC, pp 272–278
30. Grauman K, Betke M, Gips J, Bradski GR (2001) Communication via eye blinks – detection and duration analysis in real time. In: Abstracts of the IEEE computer vision and pattern recognition conference, vol 2, Kauai, HI, pp 1010–1017
31. Haro A, Flickner M, Essa I (June 2000) Detecting and tracking eyes by using their physiological properties, dynamics, and appearance. In: Abstracts of the IEEE conference on computer vision and pattern recognition, Hilton Head Island, SC
32. Hutchinson T, White Jr KP, Martin WN, Reichert KC, Frey LA (1989) Human-computer interaction using eye-gaze input. IEEE Transactions on systems, man and cybernetics, 19(6):1527–1533
33. Jacob RJK (1991) The use of eye movements in human-computer interaction techniques: what you look at is what you get. ACM Transactions on information systems, 9(3):152–169
34. Jacob RJK (1993) What you look at is what you get. Computer 26(7):65–66
35. Kapoor A, Picard RW (2002) Real-time, fully automatic upper facial feature tracking. In: Abstracts of the 5th IEEE international conference on automatic face gesture recognition, Washington, DC, pp 10–15
36. Karson CN, Berman KF, Donnelly EF, Mendelson WB, Kleinman JF, Wyatt RJ (1981) Speaking, thinking, and blinking. Psychiatry Res 5:243–246
37. Kothari R, Mitchell J (1996) Detection of eye locations in unconstrained visual images. In: Abstracts of the IEEE international conference on image processing, vol 3, Lausanne, Switzerland, pp 519–522
38. LaCascia M, Sclaroff S, Athitsos V (April 2000) Fast, reliable head tracking under varying illumination: an approach based on robust registration of texture-mapped 3D models. IEEE Transactions on pattern analysis and machine intelligence, 22(4):322–336
39. Lombardi J, Betke M (October 2002) A camera-based eyebrow tracker for hands-free computer control via a binary switch. In: Abstracts of the 7th ERCIM workshop on user interfaces for all, Paris, pp 199–200
40. Morimoto CH, Flickner M (March 2000) Real-time multiple face detection using active illumination. In: Abstracts of the 4th IEEE international conference on automatic face and gesture recognition, Grenoble, France, pp 8–13
41. Nakano T, Sugiyama K, Mizuno M, Yamamoto S (October 1998) Blink measurement by image processing and application to warning of driver's drowsiness in automobiles. In: Abstracts of the international conference on intelligent vehicles, Stuttgart, Germany. IEEE Industrial Electronics Society, pp 285–290
42. Ogawa K, Okumura T (October 1998) Development of drowsiness detection system. In: Abstracts of the 5th world congress on intelligent transport systems, Seoul
43. Open Source Computer Vision Library, Intel Corporation (2002) http://www.intel.com/research/mrl/research/opencv
44. Perkins WJ, Stenning BF (1986) Control units for operation of computers by severely physically handicapped persons. J Med Eng Technol 10(1):21–23
45. Reilly RB (September 1998) Applications of face and gesture recognition for human-computer interaction. In: Abstracts of the 6th ACM international multimedia conference on face/gesture recognition and their applications, Bristol, UK, pp 20–27
46. Rinard GA, Matteson RW, Quine RW, Tegtmeyer RS (1980) An infrared system for determining ocular position. ISA Transactions, 19(4):3–6
47. Shapiro LG, Stockman GC (2001) Computer vision. Prentice-Hall, New York
48. Singh S, Papanikolopoulos N (May 2001) Vision-based detection of driver fatigue. http://www-users.cs.umn.edu/ ∼sasingh/research
49. Stephanidis C, Savidis A (2001) Universal access in the information society: methods, tools, and interaction technologies. Universal Access Inform Soc 1(1):40–55
50. Stiefelhagen R, Yang J, Waibel A (November 2001) Estimating focus of attention based on gaze and sound. In: Abstracts of the workshop on perceptual user interfaces (PUI), Orlando, ACM Digital Library, ISBN 1-58113-448-7
51. Stringa L (1993) Eye detection for face recognition. Appl Artif Intell 7:365–382
52. Tian Y, Kanade T, Cohn J (2000) Dual-state parametric eye tracking. In: Abstracts of the 4th IEEE international conference on automatic face and gesture recognition, pp 110–115
53. Young L, Sheena D (1975) Survey of eye movement recording methods. Behav Res Meth Instrumentat 7(5):397–429
54. Zhai S, Morimoto C, Ihde S (May 1999) Manual and gaze input cascaded (MAGIC) pointing. In: Abstracts of CHI'99: ACM conference on human factors in computing systems, Pittsburgh, pp 246–253

MS ID: UAIS0062
23 May 2003 14:10 CET