Face Typing: Vision-Based Perceptual Interface for Hands-Free Text Entry with a Scrollable Virtual Keyboard

Yulia Gizatdinova Oleg Špakov Veikko Surakka
Tampere Unit for Computer-Human Interaction, School of Information Sciences,
University of Tampere, Tampere, 33014, Finland
{yulia.gizatdinova, csolsp, veikko.surakka}@sis.uta.fi

Abstract

We present a novel vision-based perceptual user interface for hands-free text entry that utilizes face detection and visual gesture detection to manipulate a scrollable virtual keyboard. A thorough experimentation was undertaken to quantitatively define a performance of the interface in hands-free pointing, selection and scrolling tasks. The experiments were conducted with nine participants in laboratory conditions. Several face and head gestures were examined for detection robustness and user convenience. The system gave a reasonable performance in terms of high gesture detection rate and small false alarm rate. The participants reported that a new interface was easy to understand and operate. Encouraged by these results, we discuss advantages and constraints of the interface and suggest possibilities for design improvements.

1. Introduction

Hands-free text entry with an on-screen virtual keyboard has been long possible using eye tracking technology: gaze here serves as a mean of pointing and key selection is typically done by dwelling the gaze on a key for about 450-1000 ms [11,1]. To speed up the interaction, short dwell times can be used; however, this may lead to so called Midas touch problem [10] when everything a user is looking at becomes selected. Thus, some interactive elements may become unintentionally selected when a user, e.g. investigates the interface. Inherit usability problems of voluntary gaze input for control-demanding tasks [9,10], high cost of commercial eye trackers, insufficient accuracy of cheap solutions and other issues, e.g. need for eye tracker (re)calibration and restriction of head movements have led to search for alternative yet accurate, fast and convenient hands-free pointing and selection methods.

Because visual interaction is natural for humans and many face and head gestures can be made on voluntary basis [15,20], computer vision is one promising technology to support vision-based perceptual user interfaces [17]. Computer vision offers non-contact and self-initialized interaction that is readily available and easy to access as opposed to other hands-free interaction methods which frequently require external equipment, e.g. eye trackers [11] or electromyography amplifiers [15]. Given a rapid progress in computer vision, hardware processing capabilities and availability of low-cost cameras, visual input becomes an important modality in hands-free text entry applications. Such systems may be especially helpful for people with motion impairments, providing easy access to computer-mediated communication and information.

1.1. Related work

A majority of the proposed vision-based interfaces provide point-only functionality by tracking face/head or facial features [14,7,12] and using the location of the tracked object as a camera mouse. Betke et al. [1] tested normalized correlation template feature tracking in a typing board application. The reported text entry speed was 31 cpm (chars per minute) when a dwell time of 0.5 s was used. Hansen et al. [9] used a marker-based head tracking for typing with a dwell-based dynamic typing application. The reported speed of communication on the first day was ~25 cpm for Danish keyboard and ~44 cpm for Japanese keyboard.

Several authors developed point-and-click visual-based interfaces which combine both camera mouse and visual gesture detection to eliminate the use of dwell time and to emulate a "single click" functionality of a computer mouse. Grauman et al. [8] utilized voluntary blinks and brow raises detected by motion analysis and normalized correlation template matching as selection gestures. The interface was tested in a letter-scanning application that required two selections to enter a single character. The typing speed (selection-only) was 5.7 cpm. Varona at al. [18] designed a system that used nose tracker to move a computer pointer and eye wink detection to execute mouse click events. The interface was applied in menu selection tasks; its text entry performance was not tested. De Silva et al. [4] applied template matching for nose tracking and a hybrid approach to detect mouth opening gesture. The interface was tested in a point-only typing application Dasher and the reported typing speed for two participants was 38 cpm.

The literature analysis revealed that vision-based text entry interfaces are still rare and insufficiently studied, therefore their applicability and limitations for this task are not yet well understood. Apart from a selection of reliable computer vision methods of visual processing, a proper design of a visual gesture set and a typing application is important. The proposed point- and select-only text entry interfaces have not yet utilized computer vision capabilities to its full potential.

1.2. Contribution

In this paper, we present a novel design of the vision-based perceptual user interface for hands-free text entry with a scrollable virtual keyboard. The interface combines three different input mechanisms to manipulate the keyboard: (1) face detection to control a computer pointer, (2) face gesture detection to select keys of the keyboard and (3) face and head gesture detection to scroll rows of the keyboard. To our knowledge, this is the first attempt to utilize more than two visual gestures as simultaneously operating activation mechanisms in a typing application.

We describe an overall design of the proposed system together with a set of visual gestures which enable efficient control over the keyboard. In this study, the actual text entry was not performed. Instead, the aim was to test computer vision methods applied for face and visual gesture detection to ensure that the overall system has a guaranteed performance. We present the results of a thorough experimentation that quantitatively defined performance of the interface in hands-free pointing, selection and scrolling tasks. At the end of the paper, we discuss advantages and constraints of the system and suggest possibilities for design improvements.

2. System overview

2.1. Scrollable keyboard

Another problem in hands-free text entry with an on-screen virtual keyboard concerns a keyboard layout that typically has large buttons to eliminate inaccuracies in point detection [11]. Therefore, a large part of the computer screen is usually occupied by a keyboard, leaving only a small area to display emerging text and other interactive elements. To reduce the area occupied by a keyboard some authors [1,3,8,9] dropped the ordinal QWERTY layout and introduced specific keyboard designs. These designs may result in long learning times required for efficient use of a keyboard [11]. Špakov and Majaranta [16] introduced a scrollable virtual keyboard that is shown in Figure 1. In contrast to other keyboards with point- or/and select-functionality, the scrollable keyboard includes an additional scrolling operation (via selection of dedicated keys), as a part of the keyboard can be hidden. It has been shown [16] that increase in cognitive load related to memorizing positions of letters in hidden rows of the scrollable keyboard affects typing speed insignificantly. In this study, visual gestures eliminated the need for scrolling keys. Thus, the scrollable keyboard allowed for testing of three functions, i.e. pointing, selection and scrolling assigned to face and visual gesture detectors in simultaneous use. A well-known QWERTY layout was used to minimize learning effects associated with memorizing positions of keys in the layout.

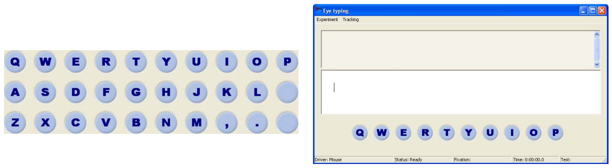

Figure 1: Left: A scrollable keyboard with all three rows visible. Right: A typing application with one-row scrollable keyboard. Text box can be expanded to allow for typing long texts.

2.2. Visual gestures

The main motivation in designing a set of visual gestures was to achieve a balance between: (1) detectability of gestures by computer vision methods, (2) elimination of fatiguing effects of visual gestures in their intensive and continuous use (3) unlikeness of gestures to occur accidentally. Therefore, a design of the visual gesture set was specified by the ease and convenience of producing these gestures on voluntary basis and discriminating visual appearance of gestures for detection purposes. Three face gestures were empirically chosen for key selection: *mouth open, brows up* and *brows down.* Two bi-directional gestures were empirically chosen for row scrolling: *brows up/down* and *head up/down.*

2.3. Camera mouse

The face detector based on a cascade of boosted classifiers with Haar-like features [19] is a real-time and robust algorithm in a variety of conditions including moderate change in illumination and head rotations. For any frame *t,* a center of the facial area $f_t=(x_t,y_t)$ detected from a video frame was mapped to a position p_t of a computer pointer on the screen:

$$p_t = \frac{1}{s}\cdot\sum_{n=0}^{s}\left(\alpha\cdot f_{t-n}+h\right) \tag{1}$$

The parameter α translates the image coordinates into the screen coordinates. It also defines horizontal and vertical scaling factors used to “amplify” head movements. A scalar parameter *h* moves the pointer's system of coordinates to the middle of the keyboard when a pointer appears on the screen for the first time. The averaging parameter *s* is used to eliminate tremors in the detected face location and provide smoother movement of the pointer. The detected facial region was further segmented using anthropometrical measures of the human face: eye-forehead area was used to detect brow gestures, mouth area was used to detect mouth openings and the whole facial region was used to detect head rotations.

2.4. Facial classifier

For face and head gesture detection a facial classifier proposed in [6,21] was utilized due to its simplicity and good classification performance. The method scales the detected region (face or feature area) to 50x50 pixels size, divides it into N=9 blocks, extracts local structural and textural features from each block and calculates a concatenated feature histogram of the entire region. The structural features are captured by the local oriented edge $LOE(\varphi_k,\sigma,K,r)$ operator [5] by convolving pixel neighborhood with a set of convolution kernels. Parameter $\varphi_k = 22.5 \cdot k$, $k = 0 \div 15$ denotes an angle of the kernel rotation, σ is a convolution coefficient, K=7 is a size of the convolution kernel and $r = 0 \div 1$ defines a resolution level. The textural features are derived from the image by local binary pattern $LBP(P,R)$ operator [13] by thresholding pixel values in a local neighborhood of P=8 points equally sampled on a circle of radius R=3. The speed of processing in case of structural features was too slow to be applied in real-time interaction context. On the contrary, the textural features were fast to extract and demonstrated good classification results. Similar results were achieved with raw intensity values (N=*1*), however, in this case training of the classifiers took much longer time than with textural or structural features.

Support vector machines [2] classifier was utilized to perform a final classification. For key selection, a 2-feature classifier *C1* was used: $C1_1 \in$ *{neutral, mouth open}*, $C1_2 \in$ *{neutral, brows up}* and $C1_3 \in$ *{neutral, brows down}*. For row scrolling, a 3-feature classifier *C2* was used: $C2_1 \in$ *{neutral, brows up, brows down}* and $C2_2 \in$ *{neutral, head up, head down}*.

3. Experimentation

3.1. Participants and apparatus

The participants were nine faculty members (25-38 years old, 4 females and 5 males) with different ethnical background, 3 had eye-glasses. Tilt-and-zoom Logitech Webcam Pro 9000 camera was placed on the top of the 17" monitor and used to achieve approximately the same face position/size in the image for each participant. Face distance from the camera was about 50 cm. The camera produced images of 320x240 pixel size and capture of 25 fps (frames per second). The illumination was kept the same for each participant. Other hardware specifications included: Intel Core 2 quad, 2.66 GHz and 3 GB RAM. The experimental software imitated the layout of the scrollable keyboard and had either 10x3 cells (full non-scrollable) layout of 1280x384 pixel size for *C1* testing, or 10x1 cells layout of 1280x128 pixel size for *C2* testing (Figure 2). A computer pointer was displayed as a green cross of 30 pixel diameter. The output window of the face processing software was visible during the experiment, so that participants were able to adjust their position in front of the camera and facilitate face detection.

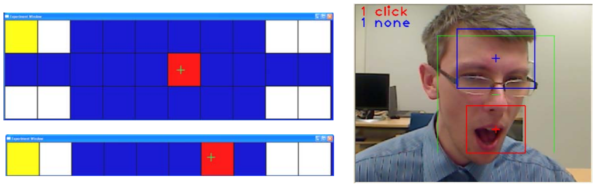

Figure 2: Left images: Experimental software that imitates 10x3 and 10x1 layouts of the scrollable keyboard. Right image: Visual output of the face processing software.

3.2. Training of the classifiers

It was expected that participants will move their heads while pointing at cells of the experimental software. Therefore, in- and out of-plane head rotations will be present. The challenge was how to train the classifiers for reliable categorization between gesture and non-gesture states. Because there are no known databases which reflect variations in facial appearance caused by these specific text entry conditions, a special procedure for collecting training data was developed. The participants were instructed to point at the top-left cell of the experimental software and to continue pointing at light cells one by one clockwise. The pointed cells were highlighted by the red color, providing visual feedback to the participants.

During this process that lasted several minutes, image data from facial areas was collected. This was done first for non-gesture and then for gesture condition. We asked participants to produce face gestures of maximum intensity in order to collect representative training sets. In average, 300 images per selection gesture, 300 images per scrolling gesture and 500 images with non-gesture condition were collected per participant. This procedure also helped the participants to get familiar with the system and facilitated development of pointing and selection strategies. For example, some participants preferred to point at the cells by using head movement and rotation, while others kept their faces frontal to the camera and moved the torso instead. The later usually resulted in a better performance of the classifiers.

3.3. Experiments

Experiment 1: The experiment started with testing *C1* classifier for making selections. During the experiment, cells of the experimental software were presented as blue and one cell (target) was randomly highlighted by the red color. The participants were instructed to point at the target and to select it using a selection gesture. After a successful completion of the task (mandatory in all trials), another randomly highlighted cell was displayed as a target. The participants needed to select all 30 cells of the 10x3 experimental software, each cell was selected once. This procedure was repeated three times, once for each selection gesture (*mouth open*, *brows up* and *brows down*).

Experiment 2: The experiment continued with testing of *C2* classifier for scrolling the rows of the 10x1 experimental software. In typing with a scrollable keyboard, users will perform scrolling in conjunction with selection operations when they point at some random key of the keyboard. Therefore, it was important to test the performance of *C2* classifier in different locations of the layout. However, the row scrolling was not performed actually. The gesture was rather treated in a way similar to "select it" that designated the end of a trial. The experimental task was similar to that in *Experiment 1*. The participants "scrolled" 10 cells in two runs: first using the "scroll up" gesture and then using "scroll down" gesture. The participants first performed face scrolling gestures (*brows up/down*) and then head scrolling gestures (*head up*/*down*).

The participants were instructed to perform pointing, selection and scrolling as fast and accurate as possible. The experiment, including training of the classifiers, lasted 45-60 minutes. A video of the participant performing the tasks was recorded as a collection of video frames processed by the system. Three input sequences, each containing approximately 730 frames, were captured per participant in *C1* testing. Four input sequences, each containing approximately 200 frames, were captured per participant in *C2* testing. For every recorded video frame, the system log file contained a time stamp, location of the target, location of the computer pointer and a class label output of the classifiers *C1* and *C2*. After the experiment, we asked participants for comments.

4. Results

4.1. Reference image data annotation

The image data collected during the experiments was manually annotated and used as a ground truth for evaluation of the classifiers. Ideally, the annotation should be repeated multiple times by different annotators in order to eliminate errors and inconsistencies from the reference data. Due to the tedious character of the work and a high number of images (over 25 000) in this study the annotation was performed by a single annotator. The analysis of the log files revealed that voluntary gestures are rather slow. The mean duration was 460 ms (SD=208) for mouth and brow gestures and 798 ms (SD=291) for head gestures. Therefore, an assumption was made that visual gestures continue over a sequence of frames and successive gestures are well separated in time. A presence of a gesture was then identified by its start-apex-end continuous succession in a sequence of frames. In this case, there was no need to identify precisely a gesture's starting and ending frames. This helped to reduce possible bias of a single-person annotation in case of transitional states which were difficult to identify either as gesture or non-gesture (see frames *t=2* and *t=5* in Figure 3). Figure 3 demonstrates a typical *mouth open* gesture. Both annotations depicted in the figure are considered correct, although in *Annotation 1* the gesture started in frame *t=3* and in *Annotation 2* the gesture started in frame *t=2*.

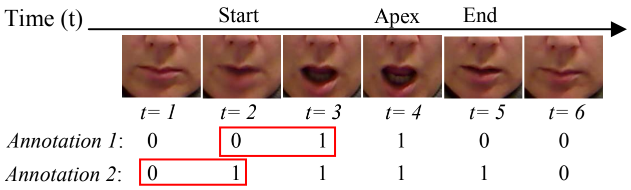

Figure 3: Two possible annotations of a sequence of video frames with *mouth open* gesture (0 and 1 define gesture and non-gesture states, correspondently).

4.2. Visual gesture detection performance

Visual gestures were detected from continuous *C1* and *C2* outputs as a transition from non-gesture state 0 to gesture state 1 (binary switch). The gesture detection was considered successful if the system triggered a binary switch close to the reference gesture location (plus/minus 1 frame), as demonstrated in Figure 3. If the system detected several gestures which corresponded to a single start-apex-end continuous succession of a gesture, all but the first were considered as false alarms. The gesture detection rate, false alarm rate and missed gesture rate were calculated as a fraction of the total number of gestures per participant. In average, gestures were correctly detected in over 95% of all cases (see Table 1).

The analysis of the log files demonstrated that gesture detectors produced multiple false alarms at the end of a gesture. To eliminate the false alarm rate, off-line filtering algorithm was applied to the system output. The algorithm made the system irresponsible for detection of fast consecutive gestures. The *Algorithm 1* utilized a threshold of about 600 ms to "freeze" the detector. The threshold was selected empirically as an optimal for a given application and a given hardware configuration. As Table 1 shows, the *Algorithm 1* resulted in considerable decrease of the false alarm rate, however, the missed gesture rate

Table 1. The performance of gesture detectors is described by average gesture detection rate *GD*, average false alarm rate *FA* and average missed gesture rate *MG*. *Alg1* and *Alg2* define less and more aggressive false alarm suppression filtering algorithms.

Classifier *C1*							
Gesture	*GD*	*FA*	*MG*	FA_{Alg1}	MG_{Alg1}	FA_{Alg2}	MG_{Alg2}
Mouth open	0.95	0.24	0.05	0.06	0.16	0.01	0.36
Brows up	1.00	0.18	0.00	0.08	0.07	0.02	0.12
Brows down	0.97	0.56	0.03	0.17	0.16	0.05	0.32

Classifier *C2*							
Gesture	*GD*	*FA*	*MG*	FA_{Alg1}	MG_{Alg1}	FA_{Alg2}	MG_{Alg2}
Head up	0.98	0.05	0.02	0.00	0.02	0.02	0.14
Head down	1.00	0.15	0.00	0.07	0.07	0.03	0.23
Brows up	0.99	0.16	0.01	0.08	0.06	0.00	0.37
Brows down	0.95	0.11	0.05	0.00	0.05	0.00	0.44

increased accordingly. After applying a more aggressive filtering *Algorithm 2*, the results became virtually false-alarm-free, but missed gesture rate increased dramatically.

It should be noted that different errors of the system will have different effect on the text entry performance. Thus, false alarms would result in unintentional entry of a character. A miss of a gesture would cause a user to repeat a gesture, resulting in slow typing speed and, possibly, frustration of a user. The *Algorithm 1* was selected to be applied in the following user studies as a reasonable compromise between the gesture detection performance and allowed misdetections of the system.

4.3. Spatial and temporal characteristics

A layout-specific analysis of the system performance is characterized by: (1) task completion time *TCT*, (2) target entry count *TEC* (defined as one plus a number of pointer re-entries to a target within a trial), (3) complete pointing time *CPT* (time interval from the target onset till the last target entry) and (4) selection time *ST=TCT-CPT* (time interval from the last target entry event till the selection event). These characteristics were analyzed relatively to: (1) distance *D* between the preceding and current target (impact on the performance of the face detector) and (2) target location on the layout (impact on the performance of the gesture detectors). The data from all tested conditions was used in this analysis. The cells of the experimental layout were squares of 128x128 pixel size. Thus, D_{MAX}*=1338* pixels for 10x3 layout, D_{MAX}=1152 pixels for 10x1 layout and D_{MIN}*=128* for both layouts. Twenty six unique distances between cells were grouped into 5 ranges (each of 250 pixels) and averages of all values falling into these ranges were computed. The lines of Figure 4 which connect these ranges for *CPT* (lower line) and *TCT* (higher line) draw a summarized effect of distance. *CPT* and *TCT* showed a clear dependence on *D* changing from 1185 to 4063 and from 1942 to 5179 milliseconds, correspondingly, as *D* increased. *TEC* also was dependent on *D* (see Figure 5). Its values averaged within each range varied from 1.16 for the shortest distances to 1.43 for the longest. The dependency of *ST* on *D* was not analyzed, as *ST* purely describes the performance of the gesture detectors and not the face detector.

The dependency of the aforementioned variables on the target location is presented next. The increase of *CPT* for targets located far from the center of the layout in most cases can be explained by the increase of the average distance that a pointer had to travel to hit a target (see Figure 6). However, *ST* and *CPT* values for the extreme cell locations in the third row (columns 1 and 10 in *Experiment 1*) are noticeably greater than those for the nearest values. The *ST* value for the left-bottom cell is a clear outlier (Grubb's test is 2.6; *Z=2.29*, *p<0.05*) and the performance of the gesture detectors can be treated as problematic in this location. A deeper analysis revealed that it was true only for *mouth open* (*SL=3366* ms, Grubbs test is 4.5; *Z=2.9*, *p<0.05*) and *brows up* (*SL=4637* ms, Grubbs test is 3.1; *Z=2.9*, *p<0.05*) gestures. For *brows down* gesture the *SL* of this cell was close to the average.

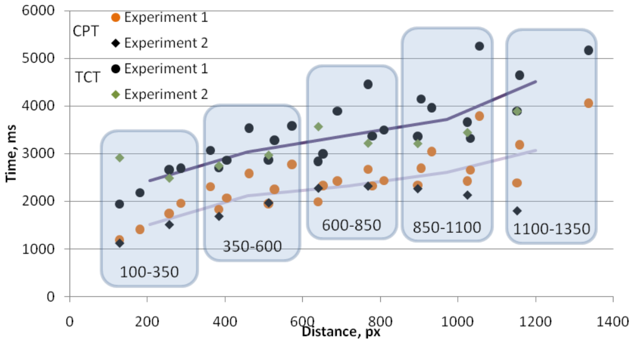

Figure 4: Complete pointing time (CPT) and task completion time (TCT) averaged over distances between cells of the experimental layout.

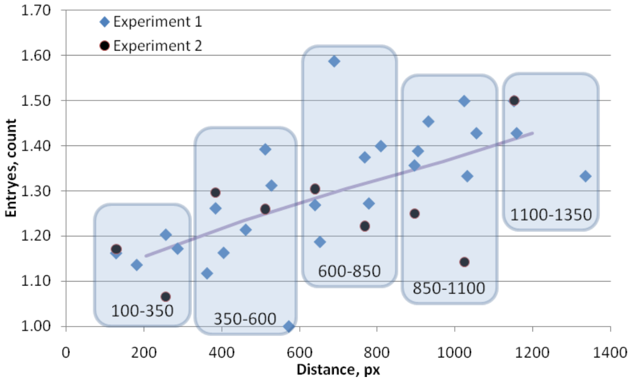

Figure 5: Target entry count averaged over distances between cells of the experimental layout.

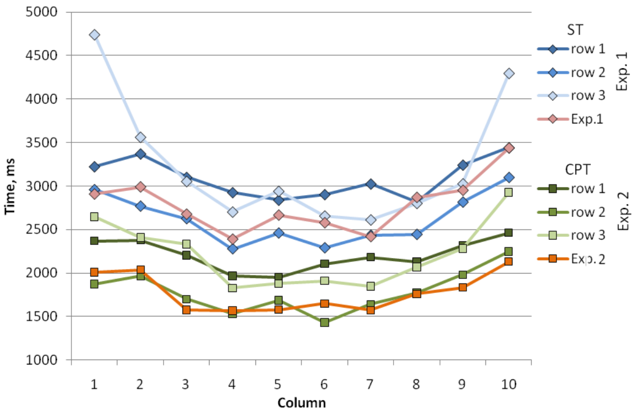

Figure 6: Complete pointing time (CPT) and selection time (ST) averaged over columns of the experimental layout.

On the other hand, the selection of cells located in the middle row was faster (2616 ms) than of the cells on other rows (3090 ms and 3239 ms; pair-wise t-test of *TCT*, *p<0.05*) in *Experiment 1*. The analysis of *CPT* also revealed similar statistics. The dependency of *TEC* on a target location also was detected: 1.52 for the left-bottom key versus the average 1.27, but it was not recognized as the outlier (Grubbs test is 2.3; *Z=2.9*). The average values of *TEC* for each row were about equal (~1.26); the dispersion of values averaged by columns was higher with the tendency to increate towards extreme columns, although without outlying values.

During the experiment, the average speed of the classifiers *C1* and *C2* was about 10 FPS. The average speed of classifiers *C1* and *C2* working simultaneously was about 8 FPS. Although the system supports real-time interaction, saving images to the computer hard drive slowed down the speed of processing.

5. Discussion and future work

We presented a novel design of the vision-based perceptual interface for hands-free text entry with the virtual scrollable keyboard. The design combined face detection for pointing at the keys of the keyboard and visual gesture detection for selecting the keys and scrolling the rows of the keyboard. To our knowledge, this is the first attempt that combined more than two visual gestures as activation input in a typing application. Both activation commands, i.e. selection and scrolling, can happen simultaneously.

We implemented the first prototype of the proposed interface and performed its proof-of-concept empirical verification. The experimental software imitated the layout of the scrollable keyboard and was used to test applicability of the chosen computer vision methods in controlling the keyboard via pointing, selection and scrolling operations. This was a necessary step prior to the future user tests in order to establish an overall performance of the system. The tested gesture classification scheme gave a reasonable performance that is compared to the state-of-the-art results [e.g. 20] and is considered acceptable for the text entry task.

The precision of the face detector was sufficient for moving a computer pointer to a desired position. For longer distances, *CPT* in Figure 4 was in average slightly slower in *Experiment 1* than in *Experiment 2*. We noticed that in pointing with 10x1 layout it was more convenient for participants to make long jumps from one cell to another. In case of pointing with 10x3 layout, a refinement of head movement in vertical direction was needed which likely caused longer times to select a target. Based on these considerations, it can be concluded that one-row scrollable keyboard design is more convenient to operate by head movements. The results from Figure 6 identified cells located far from the center of the layout as challenging to be pointed at by head movements and selected by visual gestures. Most probably this was due to the fact that participants exhibited strong head rotations (sometimes to profile views) trying to “reach” those cells. Positioning of the camera at approximately participant’s eye level helped to improve performance of the classifiers in many cases.

Due to differences in facial muscle control, the range of appearances of face gestures differed greatly among the participants. For some participants, brow gestures resulted in slight skin displacements with no wrinkles visible, which affected the detection performance of the system. This was the case, e.g. with participant 5 for whom the system produced false alarm rate of 1.96 in *brows down* selection test (and was reduced to 0.36 by applying *Algorithm 1*). In general, the system produced a large number of false alarms for this participant also for other gestures, most probably due to a failure in the training procedure. Otherwise, visual gestures were in average reliably detected across the test group. The presence of eye-glasses did not result in any noticeable changes in the performance of the system.

Practicing with the system seemed to improve the input speed for many participants. Although the text entry speed was not tested in this study, the average task completion times from Figure 4 estimate a typing speed of ~20 cpm which is comparable or superior to the results reported in the literature [1,4,8,9]. It is important to note, that whereas it may never be possible to reach a speed of a mouse/hand input (~200 cpm) with vision-based text entry interfaces, they offer feasible and available alternative for disabled users to communicate with computers (and other people through these machines). Future user studies will reveal the effects of a simultaneous use of the selection and scrolling gestures on the text entry speed.

A group of nine people tested the functionality of the system and quickly learned to use it. The participants reported that a new vision-based interface is easy to understand and operate. The system's performance can be reproduced as long as a user understands the constraints of the system and cooperates accordingly. In its present state, if a user rotates the head considerably, i.e. to profile views, the face detector fails to detect a face and the system stops working. The same happens if a user moves out of the camera's field of view. In such cases, the system could produce audio signal to inform a user that face/head position needs some adjustment in the camera view.

Encouraged by the obtained results, we plan to extend this work in several ways. Firstly, the robustness and speed of computer vision methods can be improved. As such, the design of the proposed vision-based interface is independent from its implementation, therefore, other computer vision methods than the used ones can perform better in the proposed design. For example, the use of a visual gesture classifier that does not require initial training procedure will make the system person independent.

Secondly, the results of the experiment suggested that individual differences and abilities in visual interaction varied between participants. The same gesture produced by different participants was detected with different level of robustness: in a pair of gestures *brows up* and *brows down* only *brows up* gesture was recognized very robustly for some participants and only *brows down* for others. However, the ease of making these gestures is not equal since a range of muscle control varies widely between people. In a series of future user studies we plan to explore and test a range of visual gestures that may satisfy to different user preferences. Finally, a longitudinal study of the learning effects on the text entry performance is an evident continuation of this work.

Acknowledgements

The authors thank the Academy of Finland (project 129354) and the University of Tampere.

References

[1] M. Betke, J. Gips and P. Fleming. The Camera Mouse: Visual tracking of body features to provide computer access for people with severe disabilities. *IEEE Trans. Neural Systems and Rehabilitation Engineering*, 10(1):1–10, 2002.

[2] C.-C. Chang and C.-J. Lin. LIBSVM: A library for support vector machines, http://www.csie.ntu.edu.tw/~cjlin/libsvm, 2001

[3] R. Cloud, M. Betke and J. Gips. Experiments with a camera-based human computer interface system. In *ERCIM Workshop on User Interfaces for All*, pages 103-110, 2002.

[4] G.C. de Silva, M.J. Lyons, S. Kawato and N. Tetsutani. Human factors evaluation of a vision-based facial gesture interface. In *IEEE Comp. Vision and Pattern Recognition HCI'03*, pages 52–52, 2003.

[5] Y. Gizatdinova and V. Surakka. Feature-based detection of facial landmarks from neutral and expressive facial images. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 28(1):135-139, 2006.

[6] Y. Gizatdinova, V. Surakka, G. Zhao, E. Mäkinen and R. Raisamo. Facial expression classification based on local spatiotemporal edge and texture descriptors. In *Int. Conf. Methods and Techniques in Behavioral Research*, pages 208-211, 2010.

[7] D. Gorodnichy and G. Roth. Nouse 'use your nose as a mouse' perceptual vision technology for hands-free games and interfaces. *Image and Vision Computing*, 22(12):931–942, 2004.

[8] K. Grauman, M. Betke, J. Lombardi, J. Gips and G.R. Bradski. Communication via eye blinks and eyebrow raises: Video-based human-computer interfaces. *Universal Access in the Information Society*, 2-4, 2003.

[9] J.P. Hansen, K. Tørning, A.S. Johansen, K. Itoh, H. Aoki. Gaze typing compared with input by head and hand. In *Symposium on Eye Tracking Research & Applications*, pages 131-138, 2004.

[10] R.K. Jacob. The use of eye movements in human-computer interaction techniques: What you look at is what you get. *ACM Trans. Information Systems*, 9(3): 152-169, 1991.

[11] P. Majaranta and K-J. Räihä. Text entry by gaze: Utilizing eye-tracking. In *I.S. MacKenzie and K. Tanaka-Ishii (Eds.), Text entry systems: Mobility, accessibility, universality*, San Francisco: Morgan Kaufmann, pp. 175-187, 2007.

[12] T. Morris and V. Chauhan. Facial feature tracking for cursor control. *J. Network and Comp. Applications*, 29(1):62–80, 2006.

[13] T. Ojala, M. Pietikäinen and T. Mäenpää. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 24(7):971–987, 2002.

[14] T. Palleja, W. Rubion, M. Teixido, M. Tresanchez, A.F. del Viso, C. Rebate and J. Palacin. Using the optical flow to implement a relative virtual mouse controlled by head movements. *J. Universal Comp. Science*, 14(19):3127–3141, 2009.

[15] V. Surakka, M. Illi and P. Isokoski. Gazing and frowning as a new human-computer interaction technique. *ACM Trans. Applied Perception*, 1(1):40-56, 2004.

[16] O. Špakov and P. Majaranta. Scrollable keyboards for casual eye typing. *PsychNology J.*, 7(2):159–173, 2009.

[17] M. Turk and M. Kölsch. Perceptual interfaces. Emergi*ng Topics in Computer Vision. Ed. G. Medi*oni and S. Kang. Upper Saddle River, NJ.: Prentice Hall, 456-520, 2005.

[18] J. Varona, C. Manresa-Yee and F.J. Perales López. Hands-free vision-based interface for computer accessibility. *J. Network and Comp. Applications*, 31(4): 357-374, 2008.

[19] P. Viola and M. Jones. Robust real-time face detection. *Int. J. Comp. Vision*, 57(2):137-154, 2004.

[20] Z. Zeng, M. Pantic, G.I. Roisman, T.S. Huang. A survey of affect recognition methods: Audio, visual, and spontaneous expressions. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 31:39-58, 2009.

[21] G. Zhao, X. Huang, Y. Gizatdinova and M. Pietikäinen. Combining dynamic texture and structural features for speaker identification. In *ACM Multimedia Workshop on Multimedia in Forensics, Security and Intelligence*, pages 93-98, 2010.