CAMERA BASED HEAD-MOUSE
Optimization of Template-Based Cross-Correlation Matching

Tatiana V. Evreinova, Grigori Evreinov and Roope Raisamo
Department of Computer Sciences
FIN-33014 University of Tampere, Finland

Keywords: Video-as-input, Camera-mouse, Cross-correlation matching, Reduced spiral search, Text entry.

Abstract: There is a challenge to employ video-based input in mobile applications for access control, games and entertainment computing. However, by virtue of computational complexity, most of algorithms have a low performance and high CPU usage. This paper presents the experimental results of testing the reduced spiral search with the sparse angular sampling and adaptive search radius. We demonstrated that a reliable tracking could be provided in a wide range of lighting conditions with the relative brightness of only 16 pixels composing the grid-like template (the grid step of 10-15 pixels). Cross-correlation matching of the template was implemented in eight directions with a shift of one pixel and adaptive search radius. The algorithm was thoroughly tested and after that used in a text entry application. The mean typing speed achieved with the head tracker and on-screen keyboard was of about 6.2 wpm without prediction after 2 hours practice.

1 INTRODUCTION

The latest mobile phones, PDAs and ultra-mobile PCs are equipped with digital cameras. There is a great challenge to employ video-based input in mobile applications for access control (Face Recognition, 2005), games and entertainment computing (Ballagas, 2005), (Cantzler, 2003), (YongBo, 2005), (EyeTwig, 2005), (TrackIR, 2006). There are also multi-input techniques when video input can be combined with touch and tilt/acceleration sensors served by ultra-mobile PC. To ascertain and recognize the tracked features, most of the known algorithms are executed by employing a lot of system resources being allocated for video stream pre-processing (Comaniciu, 2002), (Bérard, 1999), (Betke, 2002), (Brunelli, 1993), (Crowley, 1995), (FaceMOUSE, 2005), (Jilin, 2005), (Kamenick, 2005), (Gorodnichy, 2004). Only a few projects (Lewis, 1995), (EyeTwig, 2005), (Kjeldsen, 2001), (Si-Cheng, 2005) have been implemented to improve the input method and optimise the algorithm to make the tracking techniques more accessible and cheap. For instance, EyeTwig head tracker has a good balance of accuracy, CPU usage and a price (EyeTwig, 2005). It was realized with the Intel Open Source Computer Vision library (Intel "OpenCV", 2006). To detect the head position, a low CPU usage was achieved by using minimal and non-specific criteria such as fast ellipse fitting. Therefore, EyeTwig tracker has low filtering efficiency to discriminate head movements against an emergence and motion of another object having a similar region and brightness in a capture window.

Nowadays, the prices of web cameras are the same as those of optical mice. The requirements for video-based input techniques are not as high as, for instance, for security applications (access control and person identification). Video input may be implemented by using various or combined algorithms such as template- (or feature-), knowledge-, and appearance-based methods. The main goal is to provide smooth tracking of the body features (e.g., a particular skin region). The coordinates of the feature location in a capture window might be transformed into a cursor position, while the tracking algorithm should support sufficient accuracy in pointing and selection of icons and other widgets in the interface independently of the lighting conditions and cluttered background.

Cross-correlation (CC) algorithm has been a conventional approach in feature detection by template matching in computer vision since the

V. Evreinova T., Evreinov G. and Raisamo R. (2007).
CAMERA BASED HEAD-MOUSE - Optimization of Template-Based Cross-Correlation Matching.
In *Proceedings of the Second International Conference on Computer Vision Theory and Applications - IU/MTSV*, pages 507-514

1960's. Several schemes to simplify and accelerate computation of the cross-correlation for image processing have been proposed, e.g., (Crowley, 1995), (Lewis, 1995). Nowadays, the method has been used in machine vision for industrial inspection (defect detection) as well as for real-time tracking of the facial landmarks (Bérard, 1999), (Betke, 2002), (Crowley, 1995), (FaceMOUSE, 2005), (Jilin, 2005). However, for real-time image feature tracking there is not an agreement on which specific parameters and a way of processing are better concerning the balance of the computational costs and a reasonable reliability. It can be shown that there is a potential to optimise CC to perform video tracking with reasonable efficiency for a variety of interaction tasks. Below we will present further improvements in the CC algorithm that can speed up template matching and free CPU resources.

2 DESIGNING THE CC METHOD

Facial landmarks can be considered as a relatively rigid surface having near-equal brightness within a small field as 0.2-0.5% of the entire face image. That is, a surface area of 12×12 pixels may not have a significant brightness gradient in many regions within a frame of 320×280 pixels. A template having a size of about 1.5-2.5% of the entire face image, that is, composed of about 1600-2300 pixels, might have many dissimilar spots.

Guided by this reasoning, we have considered optimising the size of the skin region and the number of dots (pixels) which compose the template that has to be matched and tracked. These dots have two basic parameters: the coordinates and brightness. Brightness is a more general parameter than colour that may vary significantly depending on the type of the lighting and reflection conditions. As a signature of the image region, these dots should compose an array having unique features, if it is possible. They can be selected, for instance, as a rectangular grid with a fixed step along X-axis and Y-axis presenting the unique gradient of brightness, or vice versa, the template can present a particular layout of dots with near-equal brightness. The connected dots having a similar brightness and being localized in some distances can present the unique pattern as well. On the other hand, the higher are requirements to the template detection and matching, the greater are demands to the lighting conditions.

The absolute brightness of each dot (pixel) for 8-bit grey level images can vary in a range of 0-255. While a phase of varying the brightness of the connected dots in rows and columns of the grid-like template remains almost the same at the head movements within a small area, for example, yaw at less than ±20 degrees and pitch at about ±15 degrees. It can be shown that correlation between two samples calculated on the relative brightness of their dots is the same as that which is being calculated with the absolute brightness. By the relative brightness, we assign a difference between brightness of any pixel minus the mean brightness over all pixels in the template ($B_i - B_{ave}$). Still, the correlation calculated on the relative brightness is less sensitive to changes in image intensity with different lighting conditions (Lewis, 1995).

We carried out a study with different sizes of the template (from 10×10 to 50×50 pixels) composed of different numbers of pixels (from 225 to 12). We found that the 16-pixel array being arranged as a rectangular grid within the image region with a side of 40-48 pixels gives us an opportunity to choose relatively unique areas in the human face. Such a template can be tracked efficiently, that is, template matching can occur with a correlation of about 95% with a minimum number of failed records.

Next, we have proposed to reduce the full computation of correlation (Bérard, 1999), (Betke, 2002), (Lewis, 1995) over the particular search area to the restricted field through *reduced spiral search with the sparse angular sampling* starting from the initial location where the template was stored. We did an extensive evaluation of all variables of the proposed algorithm, which might be minimized without performance degradation (Evreinova, 2006). As a criterion of the algorithm performance (and the best values), we used the radius of a search area and the time of computation needed to detect a sample which resembles the template with the maximum correlation. An assessment was made for 3, 4, 6, 8 and 16 directions (with an angular step of 120-22.5 degrees) and the different radius of a search area that varied from 1 to 40 pixels.

In particular, we found that the search area might be processed only in eight directions with the angular step of 45 degrees around the starting point and the radius of a search area can be less than 20 pixels. Thus, the search area presented the rectangle each side of which was by 40 pixels greater than a side of the template. When the sample captured regarding the starting point, for instance the top left pixel, had the highest correlation coefficient among all samples recorded within the search area, coordinates of such a point were selected as the new starting point for the next search in the new capture window.

```
to the initial location where the
template was initialised; GoTo (1).
```

Throughout the tracking, matching is always performed regarding the same template (the relative brightness of 16 pixels). Still we cannot exclude the previous location from computation as it is impossible to predict a new position of the template even within 10 ms. When all of the sample candidates within the maximum capture radius failed (step 7 in the algorithm), for instance, when the correlation coefficient was less than the predefined threshold 0.8, the search started again from the starting point where the template was stored initially. To facilitate a spatial synchronization of the tracking process the user has to follow the initial head position. In general, the failed records occur when a sample crosses the borders of the working area or/and it is impossible to calculate the correlation beyond the range of the predefined parameters. Nevertheless, a reset of the template (recalibration) was never required.

The region between or near eyebrows is considered as the best facial landmark for video tracking having a sufficient brightness gradient (Bérard, 1999). The movements of the facial landmark were recorded with the Logitech QuickCam camera from the distance of about 75 cm. Normally, the template had a size of 48×48 pixels and it was always comprised of only 16 pixels (layout 4×4, the grid step of 15 pixels). The part of 160×120 pixels of the image format 320×240 pixels was centered and zoomed accordingly. In comparison to the raster-like matching carried out throughout the overall search area with the shift of 1 pixel, the reduced spiral search with the sparse angular sampling can decrease by 8.9-2.2 times redundant data. When R=20, a ratio of a number of computations is (40×40)/(20×9)≈8.9, when R=5, a ratio is (10×10)/(5×9)≈2.2.

Finally, employing Microsoft Video for Windows (VfW) and API functions that enable an application to process video data take different time and affect the tracking performance. To grab an image from the hidden capture window we used an intermediate DIB (Device Independent Bitmap) file of the capture window 320×240 pixels stored by sending WM_CAP_FILE_SAVEDIB message. It takes of about 1.2 – 2.5ms for Intel Pentium 4 3201.0 MHz Cache 2MB and AMD Athlon Processor 858.8 MHz, Cache 256kB accordingly, and the same time is needed to convert DIB to BMP format (76kB, 320×240 pixels, black and white with 8-bit greyscale) and to load the image from the file. The template-based tracking performance depends on how the algorithm and methods are integrated with other procedures. In this case, a number of computations can be optimised as well (Lewis, 1995).

As a lighting source, we have used luminescent lamps. Besides other features, the minimum correlation threshold 0.8 makes the input technique tolerant to head rolls (tilts) in a range of about ±18 degrees. It was sufficient to support head tracking when lighting conditions varied in a wide range of brightness with a minimum of failed records.

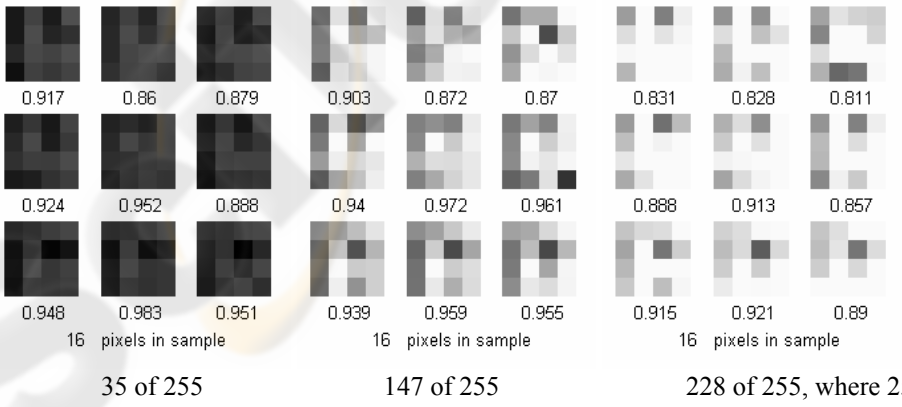

Figure 2: The variation of correlation in the target acquisition in nine locations within the working area of 80×60 pixels when brightness of the lighting source was changed gradually. The bottom figures correspond to the average brightness of the sample, the template was stored at the middle level of lighting.

Fig. 2 illustrates the algorithm tolerance to the lighting at the target (12×12 pixels) acquisition (direct pointing) in nine locations within the working area of 80×60pixels when brightness of the lighting source was changed gradually by more than six times (with the help of an electronic regulator). Each dot of the sample in Fig. 2 is shown as a square having the width and the height which are proportional to the grid step of 15 pixels in the template layout.

To support head tracking in full screen size we did a preliminary study of possible instability of the system. In particular, correlation coefficient may change due to sparse angular sampling, errors in computation (rounding), lighting and other non-controlling factors. The *system noise* had a random variation of correlation within a range of about 0.035 (STD=0.006) and an area with a radius of about one pixel (STD=0.00). The mean of Corr.max recorded was of about 0.999 (STD = 1.24). The drifting of the maximum correlation was also recorded (on 1000 counts) at neutral head position. It was of about 1.08 pixels (with a minimum of 1 pixel, a maximum of 7 pixels, STD=0.48). It can be considered as a *physiological noise* (tremor-like micromovements). Displacements of the start point may occur with equal probability in any of eight directions when no involuntary movements were produced.

Drifting of the output coordinates can be decreased through moving average procedure to stabilize displacements of the starting point ($\bar{D}$st.pt.) that should be converted into the location of the cursor ($\bar{P}$curs.) in the application program. However, an averaging on five and more pixels (including tremor and spastic movements) has a negative impact on cursor movements ("sticky" or delayed cursor etc.). To make the cursor movements smooth and to avoid the problem of the sample drifting the coordinates of the referent point were averaged on three-four locations and translated into the cursor relative displacement using power function with index K_3=2.0-2.4 (acceleration factor). The higher cursor acceleration is used the smaller head movements are needed to access any location within a screen. At a small cursor displacement, we used a threshold index K_2=0-0.35 to block acceleration of small involuntary movements (noise reduction). That demands less accuracy for the acquisition of small targets. Speed of cursor movements can also be adjusted as index K_1 =1.7-2.5. Thus, cursor displacement transfer function is calculated as a vector $\bar{P}$curs.:

$$\bar{P}\text{curs.} = \bar{P}\text{'curs.} - (K_1 \times \bar{D}\text{st.pt.} + K_2)\text{^}K_3 \qquad (1)$$

where $\bar{P}$'curs. is previous cursor position.

Such a function (1) with moving average of displacements ($\bar{D}$st.pt.) on 3-4 points requires few computations. For instance, Kjeldsen (2001) had explored application of head gestures for cursor positioning, continuous control of sound, spatial selection and symbolic selection (by yaw/no and pitch/yes). The author described a hybrid algorithm for facial pointing which take into account sensing constraints and computational efficiency. One of the key features was a sigmoid-based transfer function which combined head velocity and position to improve cursor control. Earlier Bérard (1999) also used exponential function to better adjust scrolling speed depending on head displacements.

3 EVALUATING TEXT ENTRY WITH HEAD-MOUSE

Seven volunteers from staff and students at the local university were recruited for this study. This group, which consisted of 4 females and 3 males, covered an age range from 27 to 50 years. None of the participants had any previous experience with head-mouse text entry, and three of them participated in the experiments with eye typing. Three of the subjects wore prescription glasses.

The critical parameters of software were preliminarily tested with a different CPU as mentioned in Section 2. The text entry tests were carried out on a PC with Intel Pentium 4 CPU. The monitor used 19" AL1931 ACER had a resolution of 1024 × 768 pixels. Logitech QuickCam Pro 3000 had a frame rate of 30 fps. The test program was written in Microsoft Visual Basic 6.0.

The head-mouse testing of text entry method was performed with the Standard 101-key Microsoft On-Screen Keyboard (OSK_QW) having the regular QWERTY layout. Fig. 3 illustrates a snapshot of the setup. The data were recorded after the preliminary training phase. The subjects were trained in the use of the head-mouse and text entry techniques during 4 days, but no more than (6) trials per day, 2 hours (4368 characters) in a total.

Each trial consisted of entering twenty words, randomly selected from a set of 150 words, and displayed one at a time in the test window located above the notepad (Fig. 3). The test words were 7 to 13 characters in length, with a mean of about 9, and every letter of the alphabet was included at least several times during the trial.

In order to record the time per character equally, a timer was started when the virtual space bar was pressed and stopped when a correct character was entered. When the last character of the word was entered, the test was stopped, and the next word to be entered appeared in the test window after a delay of 3 seconds. Each of the subjects accomplished eleven trials in a one-hour session.

A physical button, the Control key was used to make a selection of any software button instead of dwell-time. When the cursor position was outside of the onscreen keyboard the same key was used to start and stop the camera control.

Figure 3: A snapshot of the software used in the experimental setup: test window (top), Notepad (middle) and OSK_QW (bottom).

4 RESULTS OF EVALUATING TEXT ENTRY

Statistical data were obtained for 7 subjects entering 20 words in 11 trials, for an estimated 14014 characters in total. The key figures such as the number of errors per trial, the entry time per character and per word, and the number of keystrokes per word were stored for each trial in a data array and saved in a log file.

Fig. 4 illustrates the grand mean and standard deviation of the entry time needed to point and select any character (software button) using the head-mouse built on the improved algorithm described in Section 2 and a single physical key to make a selection.

The total summarized relative frequency of the characters used during the test is shown in the bottom of the Fig. 4. The correlation of relative frequency of the characters used during the test with English letter frequency was of about 0.985.

Fig. 5 shows the total error rate as a percentage of errors committed. It illustrates that the average error was of about 4.53% (STD = 3.57) without any prediction or text entry optimisation while making use of Microsoft on-screen keyboard. The entry of some often-used characters was less erroneous than others because the probability of their occurrences was higher, and rarely used characters have lower frequency of emerging. Therefore, the percent of error was higher due to a small number of such events.

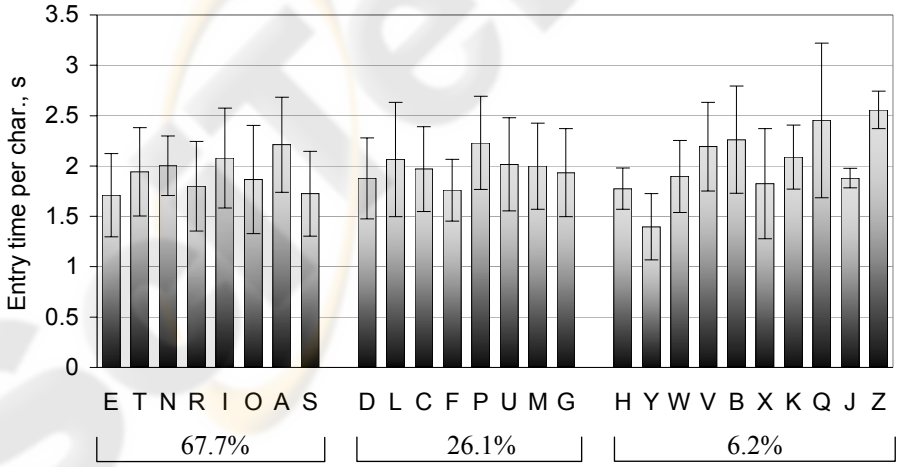

Figure 4: The grand mean for typing speed per character (and STD) with Microsoft OSK. The total summarized relative frequency of the characters used is shown in the bottom (%).

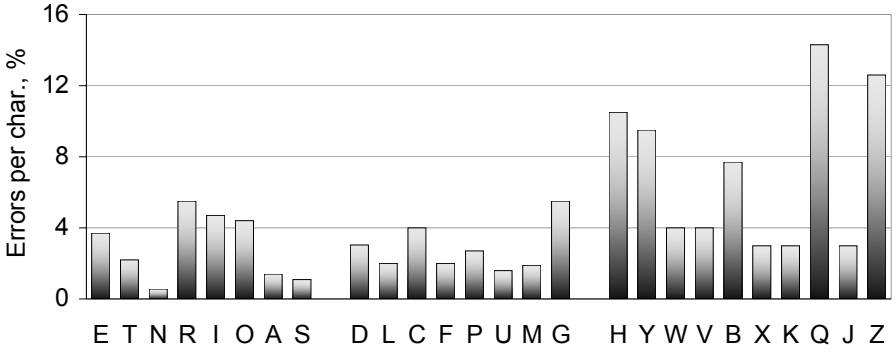

Figure 5: The total error rate summarized over 1540 words and committed by all the participants.

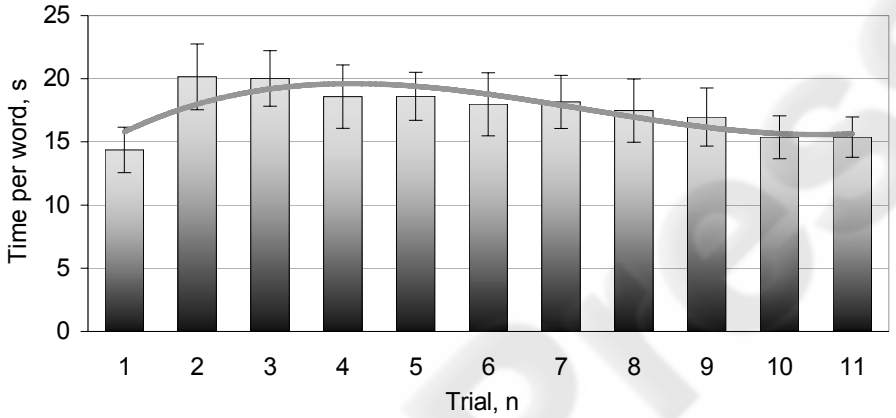

Figure 6: The grand mean for entry time of the whole words (and STD).

The average performance of the subjects can be measured as the mean time of the word entry (Fig.6). The average typing speed was of about 17.6 s (STD=9.1 s) per word (at the mean of 9.1 characters per word) that translates into 6.2 wpm (STD=3.21 wpm) based on the average length of 5 characters per word in English. The trend line indicates a normal performance variation during the test which took about one hour. In spite of the fact that nobody reported any problems with neck muscles, extreme fatigue or pain, the subjects had a rest of about 2 minutes after each trial (20 words) which could be completed in about 3 minutes. For instance, Hansen (2004) reported a typing speed of about 6.22 wpm, with the error rate of 14%, on the data being averaged on 30762 characters. The "Smart-Nav™" hands free mouse (from "Natural Point") and a dynamical Danish on-screen keyboard with word prediction/completion mode were used for testing.

Nevertheless, the size of software buttons OSK_QW keyboard was 9mm × 9mm, while the Danish keyboard occupied full screen, and the size of each button was 8×8 cm. The size of the software buttons has an impact on the usability factor such as the user satisfaction. The higher demands to head movement accuracy is the higher tension of the neck muscles could occur during voluntary head tracking. On-screen keyboard is just an input interface which should take a minimum space of the desktop.

5 CONCLUSION

We demonstrated that the information provided by the grid-like template through relative brightness of 16 pixels (layout 4×4, grid step 15 pixels) is sufficient for tracking a facial landmark in various positions at the correlation threshold more or equal to 0.8. After completing a series of extensive

experiments, we concluded that when the rectangular template has a side of 40-48 pixels, a capture radius of 20 pixels is enough to hold failed records at a minimum. Moreover, we proposed the enhanced cross-correlation algorithm with the adaptive search radius which provides the maximum degree of similarity, more than 0.95, between tracked region and the template. The matching procedure was improved and it is implemented in eight directions based on the reduced spiral search with the sparse angular sampling and shift of one pixel. Such an improvement of cross-correlation algorithm decreased the number of computations by 8.9-2.2 times in a comparison to raster-like matching when the sample candidate has to be checked throughout the overall search area with a minimum consequent displacement. The improved algorithm has a good performance employing the minimum PC resources for computation, 8-15% with Intel Pentium 4 CPU. Finally, the head-mouse application was tested with processor Pentium II 351.5MHz, Cache 512Kb, RAM 130Mb running under Windows 2000. It took of about 40-65% of the PC resources at the frame rate of 30 fps.

The tests with able-bodied participants showed the average typing speed of about 6.2 wpm with text entry technique after two-hour practice of the use of head-mouse. In the further development, we plan to increase the number of applications and features which could be adaptive.

ACKNOWLEDGEMENTS

This work was financially supported by the Academy of Finland (grant 200761 and 107278), and as a part of the project SKILLS (FP6-035005) funded by the EU Commission.

REFERENCES

Ballagas, R., Rohs, M., Sheridan J.G., 2005. Mobile Phones as Pointing Devices. In *Proc. of the Workshop on Pervasive Mobile Interaction Devices* (PERMID) at PERVASIVE 2005, Munich, Germany, 27-30.

Bérard, F., 1999. The perceptual window: Head motion as a new input stream. In *IFIP Conference on Human-Computer Interaction* (INTERACT99). IOS Press.

Betke, M., Gips, J. and Fleming, P., 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, (2002) 1-10.

Brunelli, R., Poggio, T., 1993. Face recognition: features versus templates. *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol. 15, No. 10, 1042-1052.

Cantzler, H. and Hoile, C. 2003. A novel form of a pointing device. In *Vision, Video, and Graphics*, 1-6.

Comaniciu, D., Ramesh, V., Meer P., 2002. Real-Time Tracking of Non-Rigid Objects using Mean Shift. *IEEE Conf. Computer Vision and Pattern Recognition* (CVPR'00). Vol. 2, 142-149.

Crowley, J.L., Berard, F. and Coutaz J., 1995. Finger Tracking as an Input Device for Augmented Reality. *Int. Workshop on Face and Gesture Recognition*. (IWAFGR'95), Zurich, Switzerland.

Evreinova, T.V., Evreinov G., Raisamo, R., 2006. Video as Input: Spiral Search with the Sparse Angular Sampling. In *Proc. of ISCIS 2006.* LNCS 4263, Springer-Verlag Berlin Heidelberg, 542 – 552.

Face Recognition comes to Mobile Devices, 2005, at: http://www.mobilemag.com/content/100/102/C3799/

FaceMOUSE. Product information, 2005, at: http://www.aidalabs.com/

Gorodnichy, D.O., Roth, G., 2004. Nouse 'use your nose as a mouse' perceptual vision technology for hands-free games and interfaces. Elsevier B.V. Image and Vision Computing 22, 931–942.

Hansen, J.P., Johansen, A.S., Torning, K., Kenji Itoh and Hirotaka Aoki, 2004. Gaze typing compared with input by head and hand. In *Proc. of ETRA '04, Eye Tracking and Research Applications Symposium,* ACM Press, 131-138.

Jilin Tu, T. Huang, T, Hai Tao, 2005. Face As Mouse Through Visual Face Tracking. In *Proc. of the 2nd Canadian Conf. Computer and Robot Vision*, IEEE Computer Society, 339-346.

Kamenick, T., Koenadi, A., Zhi Jun Qiu, Sze Yeung Wong, 2005. Webcam Face Tracking. CS540 Project Report, Available at: http://www.cs.wisc.edu/~jerryzhu/cs540.html

Kjeldsen, R., 2001. Head Gestures for Computer Control. In *Proc. of IEEE ICCV Workshop on Recognition, Analysis, and Tracking of Faces and Gestures in Real-Time Systems*, IEEE Computer Society, 61–68.

Lewis, J. P., 1995. Fast Template Matching. J. Vision Interface. 120-123.

Intel Image Processing library "Open CV". Website 2005 http://www.intel.com/technology/computing/opencv/overview.htm

Product information on EyeTwig.com. Website 2005. http://www.eyetwig.com

Product information on TrackIR. Website 2006. http://www.naturalpoint.com/

Si-Cheng Zhang, Zhi-Qiang Liu, 2005. A robust, real-time ellipse detector. J. Pattern Recognition 38, 273 – 287.

YongBo Gai, Hao Wang, KongQiao Wang. 2005. A Virtual Mouse System for Mobile Device. In *Proc. of the 4th international conference on Mobile and ubiquitous multimedia.* ACM Int. Conference Proceeding Series; Vol. 154 MUM, 127-131.