An Optical Pen Tracking System as Alternative Pointing Device

Ingmar Seeliger, Ulrich Schwanecke, and Peter Barth

Fachhochschule Wiesbaden - University of Applied Sciences
Kurt-Schumacher-Ring 18
65197 Wiesbaden, Germany

Abstract. A webcam together with a standard pen can replace a mouse as pointing device for many common user interaction tasks. We provide an implementation of a dedicated image-processing component integrated in a ready-to-use software tool that acts as mouse alternative. The image-processing component is responsible for tracking the head of a pen based on shape and colour information retrieved in a quick and integrated initial pen-calibration phase on the basis of a Hough transform triggered by a simple motion detection cycle. The tracked 2D position of the pen-head seen by the webcam is used to smoothly position the mouse cursor. Combined with an auto-clicking mechanism we can replace mouse-based user interaction.

The system tolerates changing lighting conditions to a certain degree, does not need time-consuming geometric camera calibration and thus works with off-the-shelf webcams. First user experiences show that this technology can partially replace mouse interaction for Repetitive Strain Injury (RSI) patients as well as completely replace mouse interaction within dedicated environments such as presentation booths or simple games.

Keywords: image processing, human computer interaction, mouse alternative, Repetitive Strain Injury (RSI)

1   Introduction

Repetitive Strain Injury (RSI) [18,5] is probably the most common occupational disease inflicted by human computer interaction [9]. Damages and injuries are mostly located in the musculoskeletal system of hand, wrist, arm, elbow, shoulder and neck resulting from thousands of repetitive motions mainly with keyboard and mouse. While ergonomically formed keyboards, trackballs, vertical mice, etc. are at least partially suitable for preventing RSI they cannot annihilate the repetitive nature of the typical work on the computer.

A promising approach is variation, which means switching among input devices while completing the daily work. While typing can often be replaced with speech recognition there are only few viable and cost-effective alternatives available that replace the mouse or mouse like devices as pointing device. We propose a simple to setup and intuitive to use alternative pointing device requiring only a classical coloured pen and a webcam as additional hardware. By moving the pen in front of the camera the user positions the mouse cursor. We concentrate on image processing capabilities that avoid

extensive and complicated camera calibration while still providing a decent recognition in most environments.

This paper is structured as follows. First, we differentiate from related work in section 2. In section 3 we provide the overall architecture and design of the pen tracking system based on the environmental assumptions and technical requirements. The initial pen calibration phase extracting colour and shape information is detailed in section 4, while actual pen tracking is handled in section 5. Using an alternative pen tracking phase based on particle filters is first motivated and then realized in section 6. We evaluate the suitability of the pen tracking system as mouse alternative in section 7 with a small group of diverse test persons and conclude in section 8.

2 Camera based Tracking

The main task in order to replace a classical pointing device lies in identifying and tracking a position of a body part such as hand, finger, head, or eye [3]. Disregarding special hardware solutions most research concentrates on processing a succession of camera images to indentify and track a position.

An example for hand tracking is the *Visual Touchpad* [13]. The input is extracted from two cameras. The user's hands move parallel to a horizontal black surface. This multitouch-interface enables for example drag and drop, selection, zooming and operating a virtual keyboard.

Finger tracking on large displays is done e.g. by *Visual Touchscreen* [4]. Here, the cursor is controlled by pointing movements with the finger. The system uses one camera and identifies the fingertip and the central point between the eyes, so that a line through fingertip and central point defines the destination on the display. Another finger tracking system is *FingerMouse* [6], a stereo camera system for wearable computing. The depth computed from the images of the two cameras helps to identify fingers as the nearest objects and then finger movement can be used for controlling a cursor.

For eye tracking typical setups need two cameras to reconstruct the pupil in 3D space [11] while there are also approaches with one camera [7]. allowing free head motion. An advantage of both systems is the absence of a complicated camera calibration.

Beside camera based tracking there are several systems that require special equipment in order to track positions, among them are multimodal interfaces [12], the wiimote [15], and step-user interfaces [14].

Our aim is to control the cursor by tracking the position of a coloured pen with a simple webcam. In contrast to finger tracking using a pen has two advantages. Firstly, tracking is much more reliable because pens have a regular shape and interference with facial skin colour is avoided. Secondly, user interaction is intuitive because explicitly grabbing an input device reflects grabbing a mouse.

3 Requirements and Design

The pen tracking system ought to be an alternative to usual mice for common daily routine work, like operating office applications, e-mail communication and Internet browsing. Based on this premise we first define the requirements and identify potential factors

influencing tracking the pen position and then describe the overall architecture of our pen tracking system.

Requirements

On the one hand, the system's resolution has to be high enough to accurately position the cursor on the control elements such as menu items and links. The replacement of the mouse has to be comprehensive and on the window system level as software will not be rewritten to take advantage of our solution.

We do not require any special surrounding but allow usage by different persons at any place and time. Thus, we require an independence of background, brightness and colour of clothes. Furthermore, users of all ages typically tremble while holding a pen up in the air. The pen tracking system has to flexibly compensate the trembling according to the individual behaviour of different users.

The acceptance of such a solution depends heavily on an easy setup. Therefore, we must refrain from a complicated camera calibration phase. The system should be cost-effective and only require a simple webcam. Therefore, we need independence of the camera resolution and must provide a working environment starting with a resolution of 320×240 pixel.

Camera and lighting

Web *cameras* differ in many parameters influencing the tracking results. The resolution determines the maximum precision of the pointing device and the focal distance defines the optimal distance of pen and camera. The images are distorted because of the bending of the optical lenses and the sensitivity of the CCD-chip (Charge Coupled Device), which is an image sensor that converts an optical image to an electric signal, determines brightness and noise in the images. Figure 1 shows some of these effects by means of three different webcams with the same position and lighting.

Fig. 1: Different cameras with the same position and lighting, (1) Unibrain Fire-i Digital Camera (Firewire), (2) Phillips FunCam (USB), (3) LogiTech QuickCam Zoom (USB)

However, the biggest influence on the images is caused by different *lighting conditions*. We may operate in an environment with artificial or natural light and have direct

and diffuse light that influence the image. Direct light can easily lead to blasting, causing loss of the important colour information. In addition to the light sources in the room, the computer monitor directly lights the user and often changes its colour depending on the background. Figure 2 shows the influences of different lightings.

Fig. 2: Different lighting on the same object (1) Natural light in the afternoon, (2) Natural light in the evening (3) Artificial light from the ceiling, bright monitor light

The effect of different *positions* of the camera with respect to the main light source are significant, as shown in figure 3. Some of the positions make it nearly impossible to detect a pen in the image. Ideally, the camera points in the opposite direction of the light source such as the room window. The user ligthing is frontal and the illumination of the room is constant. As long as the light source is not from the back, the lighting is acceptable.

Fig. 3: Different positions of the camera regarding the main light, (1) Camera points in the opposite direction, (2) Lateral position to the light, (3) Camera points to the main light

Design

The overall system design will comprise of two main parts, the image processing providing pen positions and the integration of the tracking system into a GUI toolkit. We split the image processing into two parts. One part concentrating on the initial calibrating phase where parameters are extracted that allow to identify the pen in an image. The other part concentrates on using the previously extracted parameters to provide a

continuous stream of valid positions of the pen. Both parts merit a separate section, section 4 and section 5 respectively.

For the integration of the pen tracking system into a GUI toolkit, we decided to provide a platform dependent (Windows) stand alone application, that positions the mouse cursor and emits click events through platform specific APIs. Mouse drivers typically move the cursor farther on faster mouse movement. Note that we cannot mimic such a behaviour as there is no analog to "lift the mouse and set to a new position" with a pen. Therefore, we need to assign absolute positions on screen to the positions of the pen in the camera image. To simulate clicking, we implement a simple "auto-clicker", that simulates a single left click whenever the cursor comes to a halt for a while after a movement. Other clicks and scrolling is initiated when special control keys are pressed on the keyboard simultaneously to the automated click.

In the following, we focus on the image processing features to extract valid positions and disregard implementation details regarding the window level integration of the pen tracking system.

4 Pen Calibration

In order to track the position of the pen, we first need to extract enough characteristic information to identify the pen in an image. Therefore, we extract in an initial calibration phase, where the user moves the pen once horizontal from on edge of the image to the other, four main characteristics – oblongness, movement, linear edges and monochromaticity – that serve later to identify the pen.

By *oblongness* we mean a measurement to compare different objects when more than one object is segmented as possible pen. In such cases it can serve as decision support because the pen is a very longish object.

The *movement* of the pen can be detected with help of difference-image-methods. Note, that differences with regard to a reference background are not sufficient, as the user typically moves, and if only a little, behind the pen. Thus, we compute difference-images I_t based on three consecutive images G_{t-1}, G_t, G_{t+1} with the sensitivity parameter s as follows.

$$I_t(x,y) = \begin{cases} 1 & \text{if } |G_{t+1}(x,y) - G_t(x,y)| > s \ \wedge \ |G_t(x,y) - G_{t-1}(x,y)| > s \\ 0 & \text{otherwise} \end{cases}$$

Examples are shown in figure 4. Note that this method alone is not sufficient to reliable identify pen movement, but is used for a first approximation.

Linear edges are reliably found with Hough transform [8] after creating an edge-image with a proper filter such as Laplace, Sobel, or Canny. Again, we might find not only the edges of the pen but also other edges in the image. Furthermore, the edges are not necessarily straight lines due to the bending of the camera's optical lens.

The *monochromaticity* and therefore the colour of the pen is a very important and practical information for segmentation and is the most significant characteristic feature used later in the tracking phase. The pen should be chosen in a way that it differentiates strongly from the other colours in the background. As the background and lighting

Fig. 4: Analysis of movement with difference-images, (1) With background as reference-image, (2) With three following images, movement of the body, (3) With three following images, only movement of the arm

differs, different pens might be better suited in different circumstances. Thus, the pen can be freely chosen.

Note, that the free choice of the pen and the different lightings make a calibration step before every use mandatory. In this calibration step the pen must be identified in the image and its main characteristic feature, the colour, adequately identified.

The calibration process

The result of the calibration is a colour-class with linear borders for the actual pen, which serves as basis for later tracking. We have chosen the sense-orientated HSV colour space among the colour spaces YCrCb and YUV most common in analog video technique, the device-dependend RGB and the chromatic RG colour space. HSV was the most compact representation of some experimentally created pen colour classes tolerating the most error pixels. As calibration has to be done frequently, we require only a simple calibration movement. The pen is moved vertically from right to left through the image in about two seconds. During this movement, the colour-class is calculated.

The whole process is depicted in figure 5. First, the images are preprocessed with a Gaussian filter to reduce noise. Then, the actual movement is detected with the difference-image-method based on three images. The result is the rough shape of the pen plus disturbance areas such as the moving hand or head. This is followed by a line search based on a Hough transform on the edges of the difference-images. Not only the lines are calculated, but also the line segments of the detected edges. Edges with a high deviation from the vertical are removed because of the vertical pen movement during calibration. From the remaining edge lines, in most cases just the two pen edges remain, the longest edge is interpreted as pen. The median x-value of this edge is used as actual horizontal x position of the pen. We assume to have started the calibration, if the detected edge is in the right third of the image, i.e. the pen movement has just started. If the movement does not reach the left side the calibration is cancelled and restarted. In case of a successful calibration we compute the colour-class based on the difference-images and the Gaussian filtered images. To this end, we use a fast region-labelling with an union find algorithm [2] that removes all but the biggest regions. Only the pen and – maybe connected with it – some parts of the hand remain. These images are now used for masking the Gaussian filtered images, so that the pen is successfully

segmented during movement. Finally, the parts of the hand operating the pen, which consist of all pixels below the line segments, are removed.
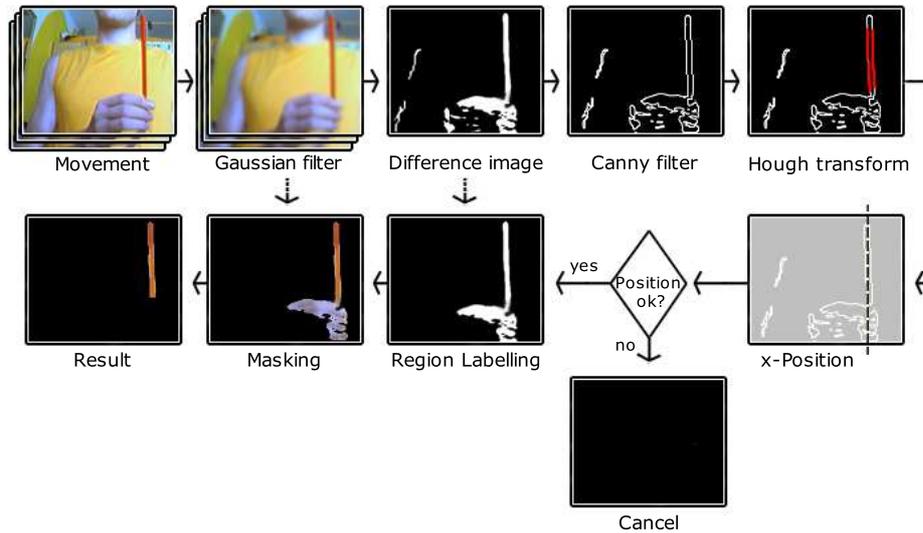
Fig. 5: Process of segmentation for pen-calibration

Figure 6 (1) shows a collage of the segmented pens during the movement. This is the basis for the creation of the colour-class. First, the image is converted in HSV-colour-space and second, a linear colour-class represented by a rectangular bounding-box is calculated which contains 90% of all non-black pixels of the image. The remaining 10% are treated as distorted noise-pixel, which result from discoloration caused by reflections especially at the border of the pen as shown in figure 6 (1). The ranges of the bounding-box in the dimensions of saturation and brightness depend on the brightness and lighting in the room. The range of hue is a static interval because of the monochromaticity of the pen, only translated depending on its actual colour. An example colour-class is shown in figure 6 (2).

We expand the colour-class to account for the discontinuous and fast changing saturation and brightness values of the image of the pen. The expansion ist limited by the first errornous background pixel falling into the colour class. This is based on the Gaussian filtered images and the fitting inverted masks.

5 Pen-Tracking

The main task while tracking the position of the pen is the *segmentation of the pen* in the stream of images from the webcam. In order to achieve fluid interaction this step needs to be fast.
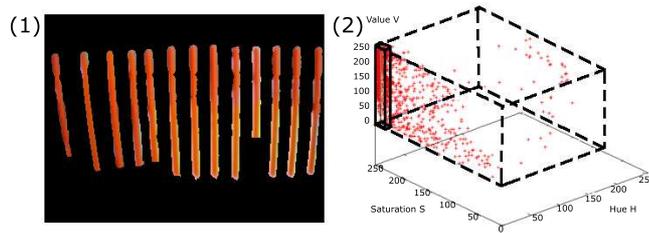
Fig. 6: Results from calibration, (1) Segmented pen during movement, (2) Colour-classes of the segmented pen. Dashed box: Class with all pixel. Solid box: Class without 10% noise-pixel

The segmentation process consists of several steps depicted in figure 7. The first step is the reduction of noise with a Gauss-Filter and conversion into HSV colour-space. Thereafter, the colour is segmented with the help of a fast colour-class-test [2]. The result is a binary image where the areas with the approximate colour of the pen are white and all others black followed by region labelling.

Next, we determine which of these regions is the searched pen by assigning probabilities of a hit to these regions and select the one with the highest probability. Note that the region with the biggest surface area is not necessarily the pen. Bigger parts of the pen are commonly not found because of light reflections which reduce the surface area of the segmentation. Therefore, we use the area only as secondary characteristic to elim-
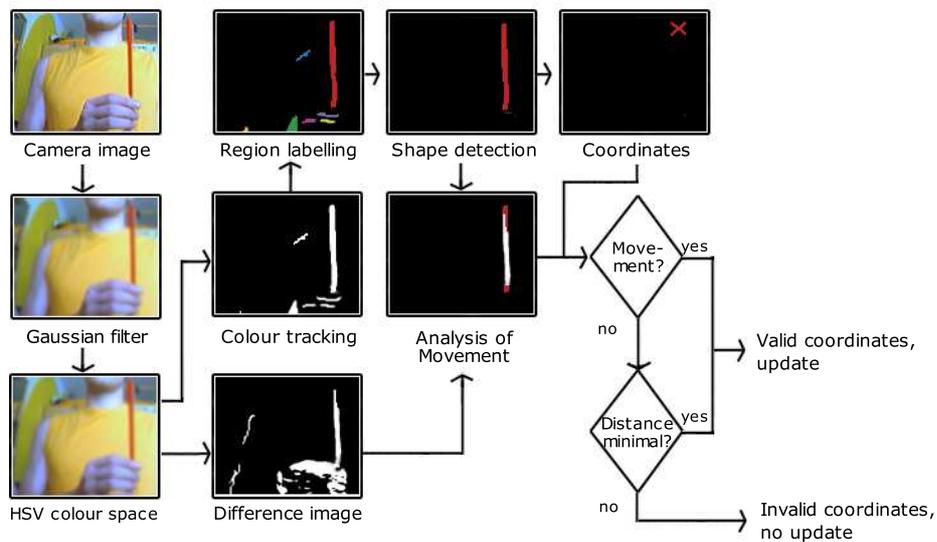
Fig. 7: The pen-tracking process

inate single error pixel and small areas. The oblongness of the pen serves as primary characteristic. It can be calculated with the help of moment-based shape features [10]. Especially the eccentricity

$$\varepsilon = \frac{(\mu_{20} - \mu_{02})^2 + 4\mu_{11}^2}{(\mu_{20} + \mu_{02})^2}$$

where μ_{20}, μ_{02} and μ_{11} are the translation-invariant centric moments of second order is a robust measure of the roundness of an object. Thereby ε various between 0 for a perfect circle and 1 for a straight line.

Finally, we use the shape to identify the pen being the most longish object.

Note that the pen is frequently put down while typing. In this case other longish segmented objects may be chosen as pen by mistake. For this reason, it is necessary to factor the actual movements of the pen into the recognition so that static background objects are not erroneously identified as pen. To this end, the difference-image is calculated and compared with the segmented pen. When a movement is discovered in this area, the found pen is valid and the coordinates of the pen are used for position the cursor. In any other case we conjecture to have either a static background object and assume that the pen is outside of the camera area, or we have the pen but it did not move sufficiently far or fast to change the cursor position. In case the distance of the actual coordinates to the last ones is small, the segmented area is in fact the pen.

Border areas

While moving the pen in front of the camera, it is inevitable that the pen is clipped at the border areas. The top of the pen will disappear at the side areas and the visible part becomes shorter, leading to distorted coordinates. The lower border area is most problematic, because the pen disappears more and more from the image the further down the pen is positioned. To accommodate for this fact, a smaller rectangle is defined inside the image as buffer for the border error. The whole image is used for tracking, but the calculated coordinates are translated into a rectangle yielding valid coordinates. Thus, we solve the problem of the border areas but decrease the available resolution.

Cursor control

The valid coordinates identified from the images are projected to the actual monitor resolution. The frame rate can be too low to use the exact coordinates as control for the cursor. Thus, it is necessary to interpolate between the coordinates to achieve a fluid movement. Additionally, a prediction filter was implemented to compensate the trembling of the hand and filter out single defective measurements. The filter predicts the next position of the pen, compares it with the actual calculated values and smoothes them. In this way, a fluid and soft cursor movement was achieved.

6 Tracking with Particle Filters

The tracking introduced in the last section does not provide accurate results if we have background areas whose colour-class fall into the colour-class of the pen as shown in

figure 8. In that case a connected area is identified during segmentation, consisting of pen and background area. This changes either the oblongness value of the segmented object so that it is not identified as pen or stretches the pen so that it is identified, but the coordinates of the top are distorted. Note that these problems can be minimized by choosing a better distinguishable pen and repositioning the camera. Still, there are
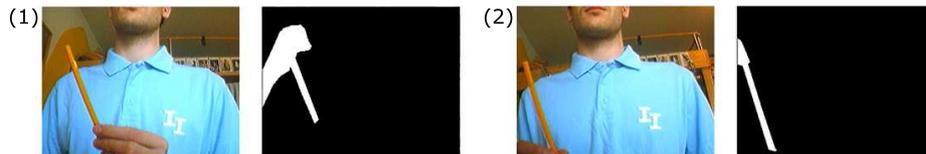
Fig. 8: Covering effects with distorted segmentation

frequent recognition errors because of light reflections on the surface of the pen. In the centre of a reflection, the brightness is 100%, the saturation is nearly zero and the hue-value varies greatly due to the blasting of the camera. In most cases these reflections result only in holes in the segmented pen, which has no consequence on pen-identification and calculation of the coordinates, because the oblongness and top of the pen are not affected. This changes, if the reflection is at the top of the pen or is so strong that it divides the segmented pen into two parts. In this case we have distortions that have to be handled.

Thus, we have implemented a *particle filter* as alternative approach to track the pen position. For the particle filter we need to identify the properties to weight the particles. Note that the oblongness of the pen, which is one of the most important properties of the last approach, cannot be used in a particle filter, because only the local environment of the particles is regarded. Therefore we use in addition to the colour the shape of the pen's top as primary property. This is the end of the pen, that is not held in the user's hand as shown in figure 9 (1), consisting of three borders which are orthogonal to each other (2). Two of the gradients point in opposite directions, whereas the third is at a right angle to both of other two. To find this special arrangement with image-processing-methods, a gradient-image is calculated with a Sobel filter as shown in figure 9 (3).

We search for the orthogonal gradients in the environment of the particles. Therefore, a gradient-histogram g which counts the relative frequency of the different gradient angles is generated for each particle. With the help of the Bhattacharyya-coefficient [1]

$$BC(p,q) = \sum_{x \in \{0,5,\ldots,360°\}} \sqrt{p(x)q(x)} \in [0,1]$$

these histograms are compared with a static reference histogram q resulting from a model of a pen top. Note that it is sufficiant to change the reference histogramm in order to recognize different shapes of pen tops.

The result is a measurement for the similarity of the compared histograms. As the value is not rotational invariant, every particle is equipped with a parameter for rotation
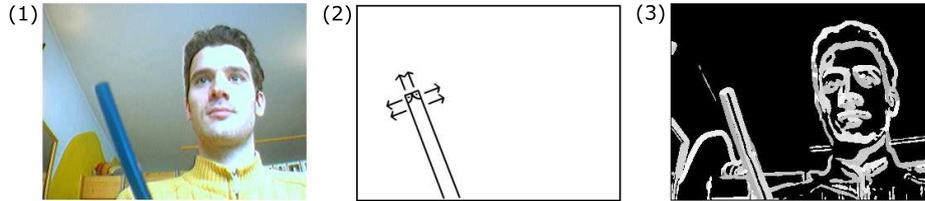
Fig. 9: Shape properties of the pen-head, (1) Original image, (2) Simplified sketch with gradients, (3) Gradient-image after use of a Sobel filter. Grey values represent the angles of the gradients.

in addition to its position. To compute rotational invariant histograms, each particle together with the associated gradients is rotated to the same vertical direction before calculating the histograms.

Among the shape of the pen's top, the colour is integrated in the weight of the particles to exclude all other areas of the image but the pen. Figure 10 shows the particle filter in action.
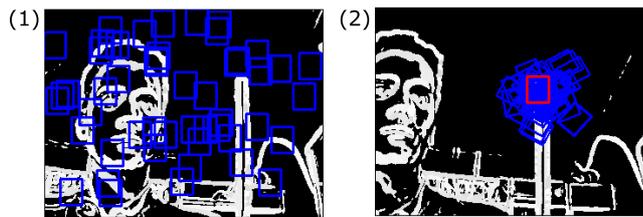
Fig. 10: Tracking with the particle filter, (1) Initial state of the particles, (2) The particles follow the pen-head

One main advantage of particle filters over other approaches is that instead of yes/no-decisions after each working-cycle probabilities that include the measurements and results of former cycles are computed. While particle filters can improve the recognition rate they cannot completely compensate for the errors caused by similar colours and reflection. Thus, both segmentation based tracking and particle filter based tracking constitute a flexible and reliable pen tracking system.

7 Evaluation

A usability test with a small group of people was conducted focusing on the applicability of the pen tracking system as alternative pointing device for RSI affected users [16]. The group consisted of 15 test persons aged 15-55 where 9 persons have no discomfort during computer work, 4 persons with some discomfort and 2 persons with obvious

RSI-symptoms as well as acute pain. They had to perform everyday tasks such as Internet research, data-handling and operating office software. The results are based on a questionnaire completed after the test by the users where the questions were graded from 1 (excellent) to 6 (unsatisfactory). Main focus was the perceived user satisfaction regarding pen-calibration, cursor movement, changing between pen and keyboard, aiming accuracy as well as learnability of the pen tracking system. The results are shown in figure 11.
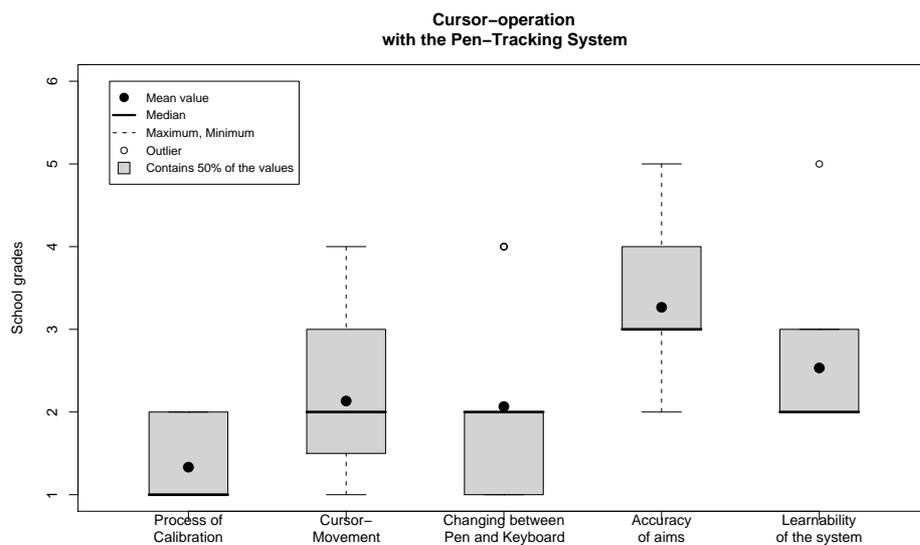
Fig. 11: Summary of cursor-operation with the pen-tracking system

In most cases, the users were satisfied with the calibration movement and also the movement of the cursor satisfied the expectations. However, the latency between user input and visible result was perceived as too high ("*The cursor operation is too sluggish.*"). The change between pen and keyboard did not pose any problems. One of the most criticized points was aiming accuracy. Opinions varied between "good" (2) and "poor" (5). The learnability of the system was classified as easy to moderate. Pointing with a pen seems to be a very intuitive gesture.

Further questions concentrate on ergonomic effects. The left diagram in figure 12 shows the result of the questions regarding strain perceived in shoulder, elbow and wrist and the right diagram shows how the strains compare to the strains while using a typical mouse. Highest strains were felt in the shoulder and the elbow. Thus, our system cannot be recommended as long term full replacement of a mouse. The range of the answers concerning strains was extremely diverse and does not allow to draw statistically significant conclusions. However, 54% of the test persons stated that in comparison to typical mice, the strain in the body seems to spread differently using our pen tracking system. This suggests that using the pen tracking device for short periods of time can temporar-

ily alleviate pain. In addition the different interaction pattern compared to mouse based interaction may help to overcome remaining nociceptive impulses that became part of the motor program initially responsible for tissue damages [17].
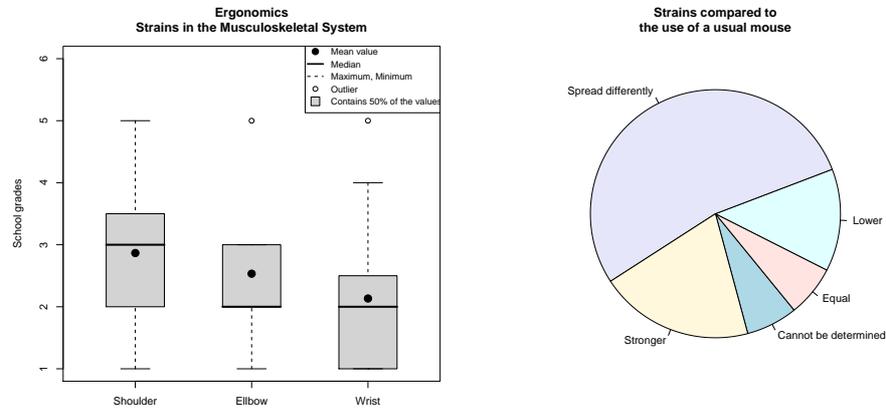
Fig. 12: Strains during the use of the pen tracking system, Left: Strains in shoulder, elbow and wrist, Right: Strains compared to the use of a usual mouse

Overall, more than half of the test group might use this kind of cursor-operation as alternative; surprisingly, also more than half of the test group without any relevant symptoms. This makes the system suitable for other scenarios such as in presentation boots and for simple games, as proposed by some test persons.

8   Conclusion

The presented pen tracking system provides an alternative input method for positioning a cursor. It employs a radically different work movement compared to the usal handling of the mouse. Furthermore, users seem to intuitively grasp the usage and accept pen based positioning as additional approach, however not as replacement.

It was possible to achieve such a solution with minimal hardware requirements using only off-the-shelf components such as a webcam and a coloured pen as well as no special environmental requirements regarding for example lighting. A demo video is available that shows the pen tracking system in action [19].

Currently we use two different approaches with different sets of characteristic properties alternatively to track the position of a pen. With just one approach it was not possible to consistently get a reliable positioning in changing environments. A joint approach combining the probabilites of the correct position has the potential to provide more stable results.

References

1. Bhattacharyya, A.: On a measure of divergence between two statistical populations defined by probability distributions. Bull. Calcutta Math. Soc., 25:99–109, 1943.
2. Bruce, J., Balch, T., Veloso, M.: Fast and Cheap Color Image Segmentation for Interactive Robots. Proceedings of IROS-2000, (2000)
3. Buxton, W.: Human Input to Computer Systems Theories, Techniques and Technology. Available at: http://www.billbuxton.com/
4. Cheng, K., Takatsuka, M.: Estimating virtual touchscreen for fingertip interaction with large displays. OZCHI '06, (2006) 97–110
5. Damany, S., Bellis, J.: It's Not Carpal Tunnel Syndrome!, RSI Theory & Therapy for Computer Professionals Simax **1** (2001)
6. Hamette, P., Tröster, G.: Architecture and applications of the FingerMouse: a smart stereo camera for wearable computing HCI. Personal Ubiquitous Comput., **12** (2008) 97–110
7. Hennessey, C. Noureddin, B., Lawrence, P.: A single camera eye-gaze tracking system with free head motion. ETRA '06, (2006) 87–94
8. Hough, P.V.C.: Method and means for recognizing complex patterns. Patent, 1962.
9. Jackson, R.: So how are your hands? Thoughts from a CS student with RSI. SIGCHI Bull, **28** (1996) 12–14
10. Jähne, B.: Digital Image Processing. Springer, edition 6, 2005.
11. Kohlbecher, S., Bardinst, S., Bartl, K., Schneider, E., Poitschke, T., Ablassmeier, M.: Calibration-free eye tracking by reconstruction of the pupil ellipse in 3D space. ETRA '08, (2008) 135–138
12. Maat, L., Pantic, M.: Gaze-X: adaptive affective multimodal interface for single-user office scenarios. ICMI '06, (2006) 171–178
13. Malik, S., Laszlo, J.: Visual touchpad: a two-handed gestural input device. ICMI '04, (2004) 289–296
14. Meyers, B., Brush, A.J.B., Drucker, S., Smith, M.A., Czerwinski, M.: Dance your work away: exploring step user interfaces. CHI '06, (2006) 387–392
15. Schlömer, T., Poppinga, B., Henze, N., Boll, S.: Gesture recognition with a Wii controller. TEI '08, (2008) 11–14
16. Seeliger, I.: Ein optisches Pen-Tracking System zur alternativen Mauszeigersteuerung für RSI-Betroffene, Diploma-Thesis, 2008
17. Sorgatz, H.: „Repetitive strain injuries" Unterarm-/Handbeschwerden aufgrund repetitiver Belastungsreaktionen des Gewebes, Orthopäde 2002 - 31, 1006–1014
18. Pascarelli, E.: Dr. Pascarelli's Complete Guide to Repetitive Strain Injury: What You Need to Know About RSI and Carpal Tunnel Syndrom. John Wiley and Sons, Inc. **1** (2004)
19. http://www.medieninf.de/pentracker