

# Collateral effects of the Kalman Filter on the Throughput of a Head-Tracker for Mobile Devices

Maria Francesca Roig-Maimó  
University of Balearic Islands  
Cra. Valldemossa, km 7.5.  
Spain 07122, Palma  
xisca.roig@uib.es

Ramon Mas-Sansó  
University of Balearic Islands  
Cra. Valldemossa, km 7.5.  
Spain 07122, Palma  
ramon.mas@uib.es

## ABSTRACT

We have developed an image-based head-tracker interface for mobile devices that uses the information of the front camera to detect and track the user's nose position and translate its movements into a pointing metaphor to the device. However, as already noted in the literature, the measurement errors of the motion tracking leads to a noticeable jittering of the perceived motion. To counterbalance this unpleasant and unwanted behavior, we have applied a Kalman filter to smooth the obtained positions. In this paper we focus on the effect that the use of a Kalman filter can have on the throughput of the interface. Throughput is the human performance measure proposed by the ISO 9241-411 for evaluating the efficiency and effectiveness of non-keyboard input devices. The softness and precision improvements that the Kalman filter infers in the tracking of the cursor are subjectively evident. However, its effects on the ISO's throughput have to be measured objectively to get an estimation of the benefits and drawbacks of applying a Kalman filter to a pointing device.

## Keywords

Kalman filter, head-tracker, throughput, Fitts' law, HCI, mobile devices.

## 1 INTRODUCTION

Head-trackers provide a hands-free way to interact with devices through the movements of the head and so, they have a direct application in assistive tools for motor-impaired users. In the assistive domain technologies, such interfaces are widely used for desktop computers [MYPVP10, MGISLVG06] and in several commercial mobile applications [DSLKT03, GB].

Research on head tracker interfaces based on image sensors for desktop computers is a mature discipline and has been conducted for a long time for HCI purposes [Toy98, BGF02, CMM<sup>+</sup>09, VMYP08]. Nevertheless, nowadays the advent of integrated frontal cameras has focused this kind of research on mobile devices.

We have developed an image-based head-tracker interface for mobile devices [RMMYV16] that only uses the information of the front camera to detect and track the user's nose position and translate its movements into a pointing metaphor to the device. However, as already

noted in the literature [CRV12], the measurement errors of the motion tracking leads to a noticeable jittering of the perceived motion. To counterbalance this unpleasant and unwanted behavior, we have applied a Kalman filter to smooth the obtained positions.

In this paper we focus on the effect that the use of a Kalman filter can have on the throughput of the developed interface. Throughput is the human performance measure proposed by the ISO 9241-411 [ISO12] for evaluating the efficiency and effectiveness of non-keyboard input devices.

The softness and precision improvements that the Kalman filter infers in the tracking of the cursor is subjectively evident. Nevertheless, its effects on the final throughput also have to be measured objectively to get an unbiased estimation of the benefits and drawbacks of applying a Kalman filter to a pointing device.

There have been some attempts to generally depict the lag that filtering inherently introduces [CRV12] but, to the best of our knowledge, there are no clues on the effects of the Kalman filter on ISO's throughput.

## 2 HEAD-TRACKER INTERFACE

FaceMe [RMMYV16] is a head-tracker interface for mobile devices that uses the information of the front camera to detect and track the user's nose position and translate its movements into interaction actions to the device (Figure 1).

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

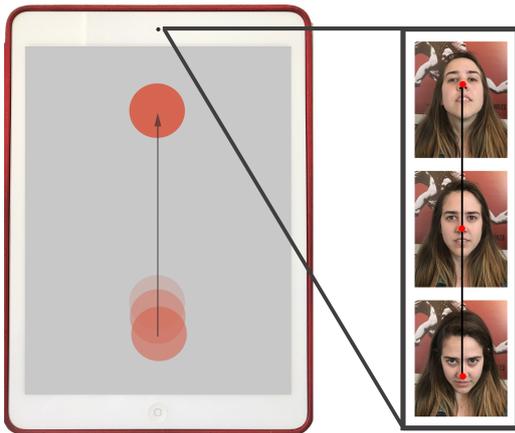


Figure 1: Example of using FaceMe as a pointing device.

A version of SINA system [VMYP08], a camera-based head-tracker interface for desktop environment, was adapted and optimized for mobile devices.

The interface is based on facial feature tracking instead of tracking the overall head or face. The selected facial feature region is the nose, because it has specific characteristics to allow tracking, it is not occluded by facial hair or glasses, and it is always visible while the user is interacting with the mobile device (even when the head is rotated).

The process is divided into two stages: the *User detection* stage and the *Tracking* stage. In the *User detection* stage we process the initial frames from the camera to detect the user's facial features to be tracked. After detection, the *Tracking* stage performs the tracking and filtering. Finally, the average of all the features (i.e., the nose point) is sent to a transfer function. This transfer function is responsible of the translation of the coordinates' change of the nose point to a coordinates' change on the device screen.

## 2.1 User Detection

In this step no calibration is needed, the only requirement is that the user must keep the head steady for a small predefined number of frames to allow the system to automatically detect the face region (see "User detected" in Figure 2).

The main face is defined as the one with the biggest area (see "Main face region" in Figure 2). To ensure a steady user for a proper algorithm initialization and to avoid false positives, we use a temporal consistency scheme (see "Temporal consistency" in Figure 2).

According to anthropometrical measurements of the human face [Sat16], the nose region occupies the second third of the facial region (see "Nose region" in Figure 3). Inside this region, the nostrils and the corners of the nose are selected as the initial facial features to track (see "Facial features" in Figure 3).

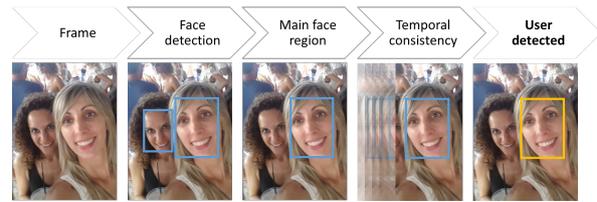


Figure 2: Illustrated theoretical stages for the detection of the main user face.

Changing light conditions can lead to the selection of unstable features, therefore we need to re-select the initial facial features using symmetry constraints (respect to the vertical axis). This leads to a more robust tracking process.

The finally chosen nose point is the average of all the facial features being tracked, which will be centered on the nose, between the nostrils (see "Nose point" in Figure 3).

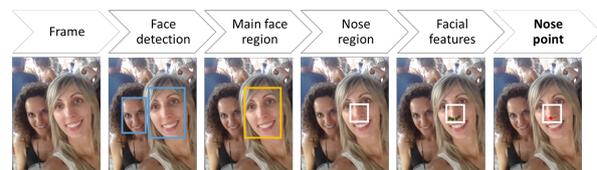


Figure 3: Simulated steps of the *User detection* stage.

The *User detection* stage works in a wide range of lighting conditions (dark or clear), users particularities (skin color, glasses or facial hair) and backgrounds (homogeneous or heterogeneous).

## 2.2 Tracking

In the *Tracking* stage, there is no need for the face to be fully visible, as only a small region surrounding the nose is used.

We get the best image registration exploring the spatial intensity gradient information of the images using a pyramidal implementation of the Lukas-Kanade algorithm [Bou01]. Since the algorithm is robust to rotation, scaling and shearing, the user can move in a flexible way. However, fast head movements can cause the lost or displacement of features to track. If we detect a feature abnormally separated from the average point, this feature is discarded (see "Filtered of displaced feature" in Figure 4). In case there are not enough features to track, the *User detection* stage restarts.

We follow a typical Bayesian approach to sensor fusion, combining measurements in the representation of a posterior probability. For each new frame, we combine the tracked nose features with newly detected features (see "Fusion" in Figure 4).

After this stage, we apply the velocity constant Kalman filter to get rid of the jittering.

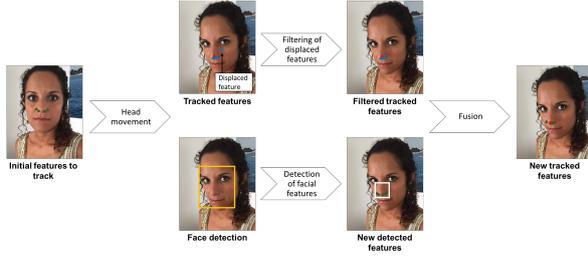


Figure 4: Simulated steps of the *Tracking* stage.

Our tracking stage is able to run in real-time on current mobile devices with a variety of CPU platforms.

A detailed description of the system is found in other sources [RMMYV16].

### 2.3 The Kalman filter

The Kalman filter is a powerful mathematical tool to be used when working with real world inaccurate measurements. It was first introduced in 1960 [Kal60] and it is still commonly used in a broad range of disciplines including satellite navigation systems [SHiS14], object and people tracking [PAHEM09] [SR11] or autonomous navigation [LFL<sup>+</sup>18].

The Kalman filter is an optimal estimation of the state of a process, in a way that minimizes the mean of the squared error. Its implementation is very fast and its memory requirements are very low, as there is no need to reprocess previously observed data.

Kalman filter algorithms work in the continuous iteration of two steps. In the first step, we update the state of our system using the dynamic model (prediction), and in the second step we update our measurement with the observation model (correction).

Our goal when using the Kalman filter is to find an estimation of the cursor position such that we obtain a smoother motion, reducing the jittering. So, in our implementation of the filter, the state of our system at time  $t$  is described with a position  $p$  and a velocity  $v$ , defining the state of the nose:

$$\bar{x}_t = (p, v)$$

The position and the velocity are correlated (the higher the velocity, the farther the motion and the slower the velocity, the nearer the motion). This correlation is described in a covariance matrix  $P_t$  where each element corresponds to the level of correlation between the couples position-velocity:

$$P_t = \begin{pmatrix} C_{pp} & C_{pv} \\ C_{vp} & C_{vv} \end{pmatrix}$$

At a time  $t$ , we need to know an estimation of the state of the system  $\hat{x}_t$ :

$$\hat{x}_t = \begin{pmatrix} p \\ v \end{pmatrix}$$

And, from the current state, we have to predict the next one  $\hat{x}_t$ .

Let us use simple kinematics:

$$p_t = p_{t-1} + \Delta t v_{t-1}$$

$$v_t = v_{t-1}$$

From which we can build a prediction matrix  $F_t$ :

$$\hat{F}_t = \begin{pmatrix} 1 & \Delta t \\ 0 & 1 \end{pmatrix},$$

such that,

$$\hat{x}_t = F_t \hat{x}_{t-1} \quad (1)$$

At time  $t$ , we also need to keep track of the covariance matrix (i.e. the prediction of the new uncertainty). We have to compute the new covariance matrix using the prediction matrix. If we multiply every element in a distribution by the prediction matrix, we get:

$$P_t = F_t P_{t-1} F_t^T$$

At this point, we can also add some additional uncertainty from the process noise expanding the covariance by adding the term  $Q_t$ :

$$P_t = F_t P_{t-1} F_t^T + Q_t \quad (2)$$

Equation 1 and Equation 2 are used to estimate the state of the system and the covariance projecting them from time  $t$  to time  $t - 1$  in the prediction step.

In the correction step, we first have to compute the Kalman gain  $K$ , using the matrix  $H$  that models the sensors relating the state with the measurements and the covariance of the observation noise  $R$ :

$$K = H_t P_t H_t^T (H_t P_t H_t^T + R_t)^{-1}$$

And now, we can state the equations for the correction step:

$$P'_t = P_t - K' H_t P_t \quad (3)$$

$$\hat{x}'_t = \hat{x}_t + K' (\bar{z}_t - H_t \hat{x}_t) \quad (4)$$

Where  $\bar{z}_t$  is the reading we have observed.

We have tuned the noise parameters so that jittering is correctly compensated in most use conditions.

Figure 5 depicts the desired trajectory, the raw data trajectory (No Kalman) and the Kalman filtering results. Although using a short path, the jittering of the red measures are clearly visible and very user noticeable in the interactive application.

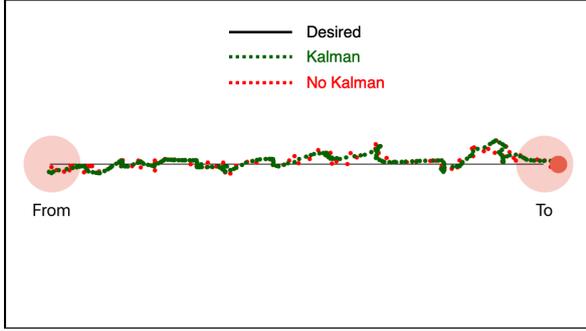


Figure 5: Desired, measured and filtered trajectories.

### 3 ISO TESTING AND THE CALCULATION OF THROUGHPUT

ISO 9241-411 [ISO12] describes performance tests for evaluating the efficiency and effectiveness of existing or new non-keyboard input devices<sup>1</sup>. The primary tests involve target-select tasks using throughput as a dependent variable.

The calculation of throughput is performed over a range of amplitudes ( $A$ ) and with a set of target widths ( $W$ ) involving tasks for which computing devices are intended to be used.

The ISO standard proposes a one-directional target-select test and a multi-directional target-select test. Due to the two-dimensional nature of the pointing metaphor, the multi-directional test is better suited for our requirements.

#### 3.1 Multi-directional Target-select Test

The multi-directional test evaluates target-select movements in different directions. The user moves the cursor across a layout circle to sequential targets of width  $W$  equally spaced around the circumference of the circle with diameter  $A$  (see Figure 6). Each sequence of trials begins and ends in the top target and alternates on targets moving across and around a layout circle.

#### 3.2 The Calculation of Throughput

The ISO standard specifies throughput ( $TP$ ) as the performance measure and it is calculated as follows:

$$TP = \frac{\text{Effective index of difficulty}}{\text{Movement time}} = \frac{ID_e}{MT}, \quad (5)$$

where  $ID_e$  is computed from the movement amplitude ( $A$ ) and target width ( $W$ ) and  $MT$  is the per-trial movement time averaged over a sequence of trials.

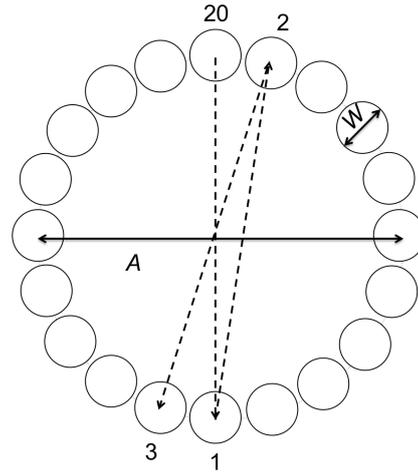


Figure 6: ISO Multi-directional target-select test.

The effective index of difficulty is a measure, in bits, of the difficulty and user precision achieved in accomplishing a task:

$$ID_e = \log_2 \left( \frac{A_e}{W_e} + 1 \right), \quad (6)$$

where  $W_e$  is the effective target width, calculated from the width of the distribution of selection coordinates made by a participant over a sequence of trials. The effective target width is calculated as follows:

$$W_e = 4.133 \cdot S_x, \quad (7)$$

where  $S_x$  is the standard deviation of the selection coordinates in the direction that movement proceeds. The effective value is used to include spatial variability in the calculation. The effective amplitude ( $A_e$ ) can also be used if there is an overall tendency to overshoot or undershoot.  $A_e$  is calculated as the mean movement distance from the start-of-movement position to the end points [SM04].

Using the effective values, throughput is a single human performance measure that embeds both the speed and accuracy in human responses. A detailed description of the calculation of throughput is found in other sources [SM04, Mac15, RMMYV17].

## 4 THE EXPERIMENT

The main goal of the experiment is to evaluate the mobile head-tracker interface following the recommendations described in the ISO standard in order to obtain a benchmark value of throughput. This will allow the comparison between the two different implementations of the head-tracker interface: by using the position obtained using the Kalman filter or by using the raw position directly.

<sup>1</sup> ISO 9241-411 [ISO12] is an updated version of ISO 9241-9 [ISO02]. With respect to performance evaluation, the two versions of the standard are the same.

## 4.1 Participants

Twelve participants (5 females) were recruited from the local town university campus in Spain. Ages ranged from 22 to 52 with a mean of 31.25 years ( $SD = 10.67$ ). There were no requirements on prior experience to participate in the study. None of the participants had previous experience with head-tracker interfaces.

## 4.2 Apparatus

The experiment was conducted on an Apple *iPad Air* with a resolution of  $2048 \times 1536$  px and a pixel density of 264 ppi. This corresponds to a resolution of  $1024 \times 768$  Apple points.<sup>2</sup> All communication with the tablet was disabled during testing.

The software implemented the ISO multi-directional target-select test (see Figure 7 for details).

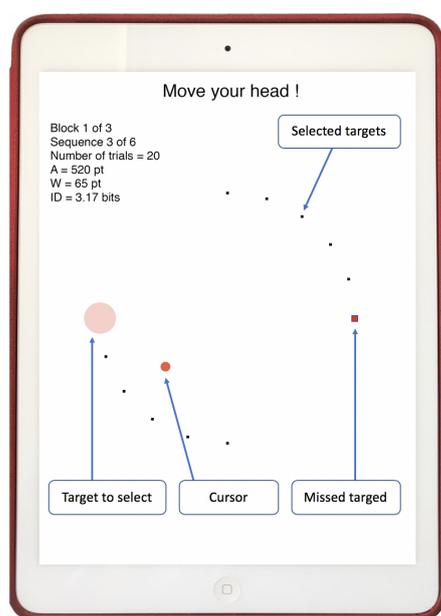


Figure 7: Screenshot of the experiment software: example target condition with annotations ( $A = 1040$  px,  $W = 260$  px).

User input combined the mobile head-tracker for pointing and touch for selection.

Each sequence consisted of 20 targets with the target to select highlighted for each trial. Upon selection, a new target was highlighted. Selections proceeded in a pattern moving across and around the layout circle until all targets were selected. If a target was missed, a small red square appeared in the center of the missed target; otherwise, a small black square appeared showing a correct selection. The target was highlighted in green when the cursor was inside it.

<sup>2</sup> Apple's point (pt.) is an abstract unit that covers two pixels on retina devices. On the *iPad Air*, one point equals 1/132 inch (Note:  $1 \text{ mm} \approx 5 \text{ pt.}$ ).

## 4.3 Procedure

After signing a consent form, participants were briefed on the goals of the experiment and were instructed to sit and hold the device in portrait orientation in a comfortable position (see Figure 8). The only requirement was that their entire face was visible by the front camera of the device.



Figure 8: Participant performing the experiment.

The experiment task was demonstrated to participants, after which they did a few practice sequences. They were instructed to move the cursor by holding the device still and moving their head. Selection occurred by tapping anywhere on the display surface with a thumb when the cursor was inside the target. Testing began after they felt comfortable with the task and the interaction method.

Participants were asked to select targets as quickly and accurately as possible and to leave errors uncorrected. They were told that missing an occasional target was OK, but that if many targets were missed, they should slow down. They were allowed to rest as needed between sequences. Testing lasted about 20 minutes per participant.

## 4.4 Design

The experiment was fully within-subjects with the following independent variables and levels:

- Filtering mode: Kalman, No Kalman.
- Block: 1, 2, 3.
- Amplitude: 260, 520, 1040 px.
- Width: 130, 260 px.

The primary independent variable was filtering mode: by applying a velocity constant *Kalman* filter to smooth the positions returned by the head-tracker interface (*Kalman filtering mode*) or by using the raw positions

directly (*No Kalman filtering mode*). Block, amplitude, and width were included to gather a sufficient quantity of data over a reasonable range of task difficulties (with *ID*s from 1.00 to 3.17 bits).

For each condition, participants performed a sequence of 20 trials. The two filtering modes were assigned using a Latin square with 6 participants per order. The amplitude and width conditions were randomized within blocks.

The dependent variables were throughput, movement time, and error rate.

The total number of trials was 12 participants  $\times$  2 interaction modes  $\times$  3 blocks  $\times$  3 amplitudes  $\times$  2 widths  $\times$  20 trials = 8,640.

## 5 RESULTS

In this section, results are given for throughput, movement time and error rate.

### 5.1 Learning Effects

Since head-tracking was unfamiliar to all participants, a learning effect was expected. Figure 9 shows the learning effect for throughput by filtering mode. The learning effect (i.e., block effect) was statistically significant ( $F_{2,22} = 11.36, p < .001$ ), confirming the expected improvement with practice. The effect was more pronounced between the 1<sup>st</sup> and 2<sup>nd</sup> blocks, with 8.65% increase in throughput, compared to a almost indiscernible decrease of 0.97% between the 2<sup>nd</sup> and 3<sup>rd</sup> blocks. A Scheffé post hoc analysis confirmed that the effect was not significant after block 1. As throughput is the dependent variable specified in ISO 9241-411, subsequent analyses are based on the pooled data from the 2<sup>nd</sup> and 3<sup>rd</sup> blocks of testing.

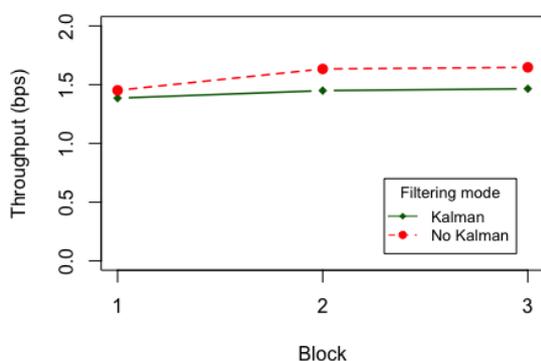


Figure 9: Results for filtering mode and block for throughput.

### 5.2 Throughput

The grand mean for throughput was 1.55 bps. This value is within the expected range for head input on mobile and desktop environments (from 1.28 bps to 2.10 bps [MFM15, DSLKT03, RMMYV18]).

The mean throughput for the *No Kalman* filtering mode was 1.58 bps, which was 10.5% higher than the mean throughput of 1.43 bps for the *Kalman* filtering mode. The difference was statistically significant ( $F_{1,11} = 7.63, p < .05$ ).

### 5.3 Movement Time

The grand mean for movement time was 1.44 s per trial. By filtering mode, the means were 1.58 s (*Kalman*) and 1.42 s (*No Kalman*). The difference was statistically significant ( $F_{1,11} = 5.92, p < .05$ ).

### 5.4 Error Rate

The grand mean for error rate was 5.58% per sequence. By filtering mode, the means were 5.70% (*Kalman*) and 5.37% (*No Kalman*). The difference was not statistically significant ( $F_{1,11} = 0.37, ns$ ).

## 6 CONCLUSION AND DISCUSSION

In this contribution, we show that to indiscriminately apply a Kalman filter to our data may lead to a decrease on the human performance in terms of the throughput of our head-tracker.

Our results show that when using the Kalman filter to smooth the positions returned by the head-tracker interface, the throughput is up to a 9.5% lower than when using the raw positions detected in the original images. Therefore, it has a negative effect on the throughput of the interface. Whether this effect is compensated by the very noticeable absence of jitter, it has to be decided depending on the application.

Results also show that although the use of the Kalman filter had no effect on the accuracy of the head-tracker in terms of error rate, it also has a significant negative effect in terms of velocity.

In the near future we are planning to evaluate the effect that some low-pass filters like the 1€ Filter [CRV12] can have on the throughput of the head-tracker used as a pointing device.

## 7 ACKNOWLEDGMENTS

This work has been partially supported by the project TIN2016-81143-R (AEI/FEDER, UE). We also thank the Balearic Islands University and its Department of Mathematics and Computer Science for their support.

## 8 REFERENCES

- [BGF02] 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 Transactions on Neural Systems and Rehabilitation Engineering*, 10(1):1–10, March 2002.
- [Bou01] J. Y. Bouguet. Pyramidal implementation of the affine lucas kanade feature tracker description of the algorithm. *Intel Corporation*, 5(1-10):4, 2001.
- [CMM<sup>+</sup>09] Fernando Caballero, Iván Maza, Roberto Molina, David Esteban, and Aníbal Ollero. A robust head tracking system based on monocular vision and planar templates. *Sensors*, 9(11):8924–8943, 2009.
- [CRV12] Géry Casiez, Nicolas Roussel, and Daniel Vogel. 1€ filter: A simple speed-based low-pass filter for noisy input in interactive systems. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '12, pages 2527–2530, New York, NY, USA, 2012. ACM.
- [DSLKT03] Gamhewage C De Silva, Michael J Lyons, Shinjiro Kawato, and Nobuji Tetsutani. Human factors evaluation of a vision-based facial gesture interface. In *Proceedings of the Computer Vision and Pattern Recognition Workshop - CVPRW 2003*, pages 52–52, New York, 2003. IEEE.
- [GB] Google and Beit Issie Shapiro. Go Ahead project.
- [ISO02] ISO. 9241–9. 2000. Ergonomics requirements for office work with visual display terminals (VDTs) – part 9: Requirements for non-keyboard input devices. *International Organization for Standardization*, 2002.
- [ISO12] ISO. 9241–411. 2012. Ergonomics of human-system interaction – part 411: Evaluation methods for the design of physical input devices. *International Organization for Standardization*, 2012.
- [Kal60] R E Kalman. A New Approach to Linear Filtering and Prediction Problems. *Journal of Basic Engineering*, 82(1):35–45, mar 1960.
- [LFL<sup>+</sup>18] Yahui Liu, Xiaoqian Fan, Chen Lv, Jian Wu, Liang Li, and Dawei Ding. An innovative information fusion method with adaptive kalman filter for integrated ins/gps navigation of autonomous vehicles. *Mechanical Systems and Signal Processing*, 100:605 – 616, 2018.
- [Mac15] I Scott MacKenzie. Fitts' throughput and the remarkable case of touch-based target selection. In *Proceedings of the 17th International Conference on Human-Computer Interaction - HCII 2015*, pages 238–249, Switzerland, 2015. Springer.
- [MFM15] John Magee, Torsten Felzer, and I Scott MacKenzie. Camera Mouse + ClickerAID: Dwell vs. single-muscle click actuation in mouse-replacement interfaces. In *Proceedings of the 17th International Conference on Human-Computer Interaction - HCII 2015*, pages 74–84, Switzerland, 2015. Springer.
- [MGiSLVG06] César Mauri, Toni Granollers i Saltiveri, Jesús Lorés Vidal, and Mabel García. Computer vision interaction for people with severe movement restrictions. *Human Technology: An Interdisciplinary Journal on Humans in ICT Environments*, 2(1):38–54, 2006.
- [MYPVP10] Cristina Manresa-Yee, Pere Ponsa, Javier Varona, and Francisco J. Perales. User experience to improve the usability of a vision-based interface. *Interacting with Computers*, 22(6):594–605, 2010.
- [PAHEM09] Saira Saleem Pathan, Ayoub Al-Hamadi, Mahmoud Elmezzain, and Bernd Michaelis. Feature-supported multi-hypothesis framework for multi-object tracking using kalman filter. In *WSCG 2009: Full Papers Proceedings: The 17th International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision*, WSCG '09, pages 197–202, University of West Bohemia, Plzen, Czech Republic, 2009.
- [RMMMYV17] Maria Francesca Roig-Maimó, I. Scott MacKenzie, Cristina Manresa-Yee, and Javier Varona. Evaluating fitts' law performance with a non-iso task. In *Proceedings of the XVIII International Conference on*

- Human Computer Interaction*, Interacción '17, pages 5:1–5:8, New York, NY, USA, 2017. ACM.
- [RMMMYV18] Maria Francesca Roig-Maimó, I. Scott MacKenzie, Cristina Manresa-Yee, and Javier Varona. Head-tracking interfaces on mobile devices: Evaluation using fitts' law and a new multi-directional corner task for small displays. *International Journal of Human-Computer Studies*, 112:1 – 15, 2018.
- [RMMYV16] Maria Francesca Roig-Maimó, Cristina Manresa-Yee, and Javier Varona. A robust camera-based interface for mobile entertainment. *Sensors*, 16(2), 2016.
- [Sat16] Robert T Sataloff. *Sataloff's Comprehensive Textbook of Otolaryngology: Head & Neck Surgery: Facial Plastic and Reconstructive Surgery*, volume 3. JP Medical Ltd, 2016.
- [SHiS14] Halil Ersin Soken, Chingiz Hajiyev, and Shin ichiro Sakai. Robust kalman filtering for small satellite attitude estimation in the presence of measurement faults. *European Journal of Control*, 20(2):64 – 72, 2014.
- [SM04] R William Soukoreff and I Scott MacKenzie. Towards a standard for pointing device evaluation: Perspectives on 27 years of Fitts' law research in HCI. *International Journal of Human-Computer Studies*, 61(6):751–789, 2004.
- [SR11] Beril Sirmacek and Peter Reinartz. Kalman filter based feature analysis for tracking people from airborne images. In *ISPRS workshop high-resolution earth imaging for geospatial information, Hannover, Germany*, 2011.
- [Toy98] Kentaro Toyama. "look, ma - no hands!" hands-free cursor control with real-time 3d face tracking. *PUI98*, 1998.
- [VMYP08] Javier Varona, Cristina Manresa-Yee, and Francisco J. Perales. Hands-free vision-based interface for computer accessibility. *Journal of Network and Computer Applications*, 31(4):357 – 374, 2008.