Probabilistic Intentionality Prediction for Target Selection Based on Partial Cursor Tracks

Bashar I. Ahmad[1], Patrick M. Langdon[2], Pete Bunch[1], and Simon J. Godsill[1]

[1] Signal Processing and Communications Laboratory
[2] Engineering Design Centre
Department of Engineering, University of Cambridge, UK
{bia23,pml24,pb404,sjg30}@cam.ac.uk

Abstract. Pointing tasks, for example to select an object in an interface, constitute a significant part of human-computer interactions. This motivated several studies into techniques that facilitate the pointing task and improve its accuracy. In this paper, we introduce a number of intentionality prediction algorithms to determine the intended target *a priori* from partial cursor tracks. They yield notable reductions in the pointing time, aid effective selection assistance routines and enhance the overall pointing accuracy. A number of benchmark prediction models are also restated within a statistical framework and their probabilistic interpretation is utilised to calculate their corresponding outcomes. The relative performance of all considered predictors is assessed for point-click task data sets pertaining to both able-bodied and impaired users. Bayesian adaptive filtering is deployed to smooth highly perturbed mouse cursor tracks that are typically produced by motor impaired users undertaking a pointing task.

Keywords: cursor movement, target assistance, intentionality prediction, Bayesian inference.

1 Introduction

With the proliferation of technological devices and their wide use in work as well as domestic environments, Human-Computer Interaction (HCI) became an integral part of modern life. Pointing at a target is a fundamental task in graphical user interfaces aimed at selecting buttons, menus, etc. Its reliability and accuracy is of a key importance for the design of effective user interfaces. This triggered an immense interest in techniques that facilitate the pointing task by reducing the cursor pointing time and improving its accuracy [1-14]. The problem is particularly challenging given the increasingly diverse population of users, for example motion impaired or able-bodied users, elderly or young users and expert or non-expert users. Accordingly, some users can find the pointing task difficult or even overwhelming at times, especially the motor impaired. In this paper, we introduce probabilistic intentionality predicators to determine in advance the intended target from partial cursor movements in a 2-D set up. The sought objective is to ease and expedite the target selection process on a computer display.

C. Stephanidis and M. Antona (Eds.): UAHCI/HCII 2014, Part III, LNCS 8515, pp. 427–438, 2014.

The characteristics of the cursor movements have been examined in several studies and there is a long history of using Fitts's Law to describe the pointing operation on a computer display and build models of targeting in HCI [1,3,4]. It stipulates that the targeting difficulty is determined by the Index of Difficulty (IoD), which is calculated based on the size of the target and its distance from the starting location. Additionally, the pointing duration can be correlated with the difficulty index. More recently, Kopper *et al* reported that the angular width of a target and the angular amplitude of the movement to the target better model the IoD [5, 6]. An easier and quicker target selection process can be achieved by deploying algorithms that can increase the target size, use larger cursor activation regions, move targets closer to the cursor location, drag cursor to the nearest target [7, 8, 9], etc. However, interactive systems typically display several selectable targets in close proximity. Their layouts have an ever increasing complexity and the targets can have varying sizes and shapes. The inability of the previously mentioned pointing assistive algorithms to determine the intended target in such typical environments was highlighted in [7] as one of their key limitations. It is noted that any erroneous selection can demand additional cognitive as well as movements abilities which can be overwhelming for some users.

As an alternative, researchers have been exploring algorithms that reduce the pointing time and facilitate the selection process by dynamically predicting the intended target on the screen from partial pointing tracks. One of the first target prediction algorithms was proposed by Murata [10], it is dubbed the Bearing Angle (BA) technique. It is based on the premise that the selectable target with the minimum accumulative angle deviation with respect to the partial cursor trajectory is the intended target. It was noted in [11] that BA performs poorly if more than one target is present in the cursor direction of travel, particularly when the cursor is far from the cluster of nominal targets. Previous results on the kinematics of pointing tasks were applied in [11] to show that the cursor movement peak velocity and the distance to the target are linearly related; the destination is accordingly predicted using linear regression. A more complex motion kinematics technique was proposed in [12] assuming a minimum jerk law for pointing motion and fitting a quadratic function to partial trajectory to predict the endpoint(s). However, the cursor tracks for motor impaired users are highly nonlinear since they experience tremor, muscular spasms and weakness [13]. The trajectories exhibit a high level of perturbations with several stops and erratic jumps in rather random directions. This renders the regression-based approaches ineffective for motor impaired users. In [14], a target predictor that is based on inverse optimal control within a machine learning framework was introduced. It leverages the maximum entropy variant to obtain the probabilities of the selectable target from a partial cursor trajectory via Bayes' rule. The inverse-optimal-control method has a high computational cost compared to the considered methods here. It requires a substantial parameter training routine, e.g. learning the state-action costs, and imposes stringent constraints on the trajectories dynamics.

In this paper, we evaluate a number of probabilistic intentionality prediction algorithms that are characterized by simplicity and low computational complexity. They deliver notable improvements to the pointing process by predicting the correct target from a small number of cursor movement points, for example 20% of the cursor track can suffice to make a correct prediction on the intended target. Bayesian state space filtering, namely Linear Kalman Filtering (LKF), is also deployed to smooth anomalous cursor trajectories. Thus, for users with motor impairments, the substantial achieved reduction in the difficulty level of the pointing task can render an otherwise inaccessible applications accessible. Even small improvements on the efficiency of the target selection process, e.g. saving few milliseconds, can have significant aggregate benefits given the prevalence of interactions through graphical user interfaces.

The rest of the paper is organized as follows. In Section 2, the tackled problem is formulated and the considered probabilistic prediction framework is outlined. In Section 3, a number of prediction models are described and their trade-offs highlighted. They are subsequently tested in Section 4 and conclusions are drawn in Section 5.

2 Problem Formulation and Adopted Approach

The tackled problem is predicting the intended target out of a set of N possible ones $\{B_i\colon\ i = 1,2,\dots,N\}$, from a partial cursor movement track $\mathbf{c}_{1:k}$. The latter is defined by $\mathbf{c}_{1:k} \triangleq \{\mathbf{c}_1, \mathbf{c}_2, \dots, \mathbf{c}_k\}$ where $\mathbf{c}_n = [x_{t_n} \quad y_{t_n}]^T$ denotes the recorded cursor coordinates along the x and y axes at time instant t_n; $\mathbf{x}^T$ is the transpose operation. Whereas, t_M is the total time duration it takes the user to select an item on the screen (for example starting at time $t_0 = 0$ for simplicity) and the full cursor track is $\mathbf{c}_{1:M}$. The locations of the selectable targets in the interface are known *a priori* where $\mathbf{b}_i = [b_{x_i} \quad b_{y_i}]^T$ is the position of the "i^{th}" button. We note that the term target and button are used interchangeably in the reminder of the paper.

The intentionality prediction problem is equivalent to calculating the maximum likelihood or Maximum a Posteriori (MAP) for the set of N possible selectable buttons from $\mathbf{c}_{1:k}$. It can be stated as

$$i^*(t_k) = \underset{i=1,2,\dots,N}{\arg\max}\ P(B_i|\mathbf{c}_{1:k}) \tag{1}$$

and $B_{i^*}(t_k)$ is the decided target at time t_k. Let t_C be the time instant at which the intentionality prediction algorithm reaches a correct decision, i.e. $B_{i^*}(t_k)$ is the correct target such that $t_C \le t_M$.

Following (1) and using Bayes' rule the objective becomes calculating

$$P(B_i|\mathbf{c}_{1:k}) \propto P(B_i)P(\mathbf{c}_{1:k}|B_i) \tag{2}$$

for each of the selectable buttons. Assuming a uniform prior on all buttons, i.e. $P(B_i) = 1/N$ for $i = 1,2,\dots,N$, determining (2) and thereby (1) depends solely on the likelihood probability $P(\mathbf{c}_{1:k}|B_i)$. In a general set up, distinct or weighted probabilities can be allocated to each of the target buttons, e.g. based on the buttons layout or the user profile.

Algorithm 1. Probabilistic MAP Estimator

Input: A partial cursor trajectory at time $\{\mathbf{c}_{k-L}, \mathbf{c}_{k-L+1}, \dots, \mathbf{c}_k\}$
Output: Intended target $B_{i^*}(t_k)$
1. Smooth the anomalies in the last logged cursor trajectory; $\hat{\mathbf{c}}_k = \mathcal{F}(\mathbf{c}_k)$
2. Calculate the likelihood probability $P(\hat{\mathbf{c}}_{k-L:k}|B_i)$ for $i = 1,2,\dots,N$ given a chosen prediction model.
3. Determine the posterior distribution $P(B_i|\hat{\mathbf{c}}_{k-L:k})$ of the N selectable targets.
4. Make a MAP choice using (1).

In a given experiment, it might be desirable to utilise only the last L cursor positions, i.e. $\mathbf{c}_{k-L:k} \triangleq \{\mathbf{c}_{k-L}, \mathbf{c}_{k-L+1}, \dots, \mathbf{c}_k\}$ and $k - L > 0$, to determine $B_{i^*}(t_k)$. A sliding time window is applied to the data and the window width is a design parameter. The adopted prediction approach at time instant t_k is depicted in Algorithm 1. It gives a generic framework encompassing the set of addressed predictors that model $P(\mathbf{c}_{1:k}|B_i)$ in Section 3.

It is noted that a practical intentionality predictor should satisfy the following important requirements [14]:

- **Efficiency:** low complexity makes the algorithm amenable to a real-time implementation. This is a critical factor for facilitating the pointing task in graphical user interfaces which are typically completed within a fraction of a second. Off-line computationally intensive algorithms that introduce high delays are not practical.
- **Case independent:** the technique should be independent of the application, selections sequence, target layouts, etc. This is due to the fact that interfaces may significantly vary between different applications and contexts.
- **Adaptability:** the nature of the pointing trajectory is greatly affected by the physical ability of the user, the input device accuracy, level of expertise or experience, etc. An intentionality predictor should be able to take such user capabilities into account.

As it will be apparent in the following sections, the adopted approach and all its associated algorithms fulfil the above requirements. Calculating the posterior probabilities for all the listed algorithms is straightforward and can be case independent. Additionally, the level of performed data smoothing/filtering can be adapted to the user abilities and the level of perturbations in the input data.

Henceforth, various algorithms that allow calculating $P(B_i|\mathbf{c}_{1:k})$ are tested on cursor data collected for able and impaired users. The performance of these algorithms is measured in terms of the percentage of time during which the correct button is chosen by the applied predictor; the saving in the pointing time or duration is $t_M - t_C$.

3 Intentionality Prediction Algorithms

Below, a number of algorithms that enable a MAP decision using (1) are described. The objective is achieving performance gains whilst maintaining simplicity and low computational complexity.

3.1 Nearest Neighbour (NN)

This is a simple and intuitive model that relies on selecting the button that is closest to the current cursor position. It relies on measuring the distance between the position of the nominal buttons $\{\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_N\}$ and $\mathbf{c}_k$ at time t_k. It allocates the highest probability to the button with the smallest Euclidian distance $\|\mathbf{b}_i - \mathbf{c}_k\|_2$. In a probabilistic framework, this can be expressed as

$$P(\mathbf{c}_k|B_i) = \mathcal{N}(\mathbf{c}_k|\mathbf{b}_i, \boldsymbol{\sigma}_{NN}^2) \tag{3}$$

where $\boldsymbol{\sigma}_{NN}^2$ is the covariance matrix. The observation vector $\mathbf{c}_k$ has a normal distribution with a mean equal to that of the selectable target location in question and a fixed variance whose value is a design parameter. Assuming that the cursor movements at various time instants are independent for simplicity, we reach

$$P(\mathbf{c}_{k-L:k}|B_i) = \prod_{n=k-L}^{k} P(\mathbf{c}_n|B_i). \tag{4}$$

For an equal prior on all the buttons, i.e. $P(B_i) = 1/N$, then (4) suffices to determine the intentionality outcome as per (1) and (2); movements along the *x* and *y* axes are reasonably assumed to be independent. It is noted here that the choice of $\boldsymbol{\sigma}_{NN}^2$ does not alter the MAP outcome.

3.2 Bearing Angle (BA)

This algorithm is based on the fact that as the cursor is heading towards the target button, the cumulative angle between the direction of travel and the position of the target is minimal [10]. The bearing angle from two consecutive cursor positions with respect to a target can be assumed to be a random variable with zero mean and fixed variance. Hence we can write

$$P(\mathbf{c}_k|\mathbf{c}_{k-1}, B_i) = \mathcal{N}(\theta_{i,k}|0, \boldsymbol{\sigma}_{BA}^2) \tag{5}$$

where $\theta_{i,k} = \angle(\mathbf{v}_k, \mathbf{b}_i - \mathbf{c}_k)$, $\mathbf{v}_k = \mathbf{c}_k - \mathbf{c}_{k-1}$ is the velocity or heading vector. Operator $\angle(\mathbf{a}, \mathbf{b})$ returns the angle between the vectors $\mathbf{a}$ from $\mathbf{b}$ using the dot product definition in a Euclidean space. Equation (5) stipulates that a smaller $\theta_{i,k}$ implies that the "i^{th}" target is more probable; this reflects the rationale behind BA. It follows that

$$P(B_i|\mathbf{c}_{k-L:k}) = \frac{P(B_i)P(\mathbf{c}_{k-L}|B_i)\prod_{n=k-L+1}^{k} P(\mathbf{c}_n|\mathbf{c}_{n-1}, B_i)}{P(\mathbf{c}_{k-L:k})} \tag{6}$$

which incorporates the cumulative sum of the bearing angle depending on the chosen width of the applied time window.

The probabilistic interpretation of BA illustrates that a confidence interval of a width set by $\boldsymbol{\sigma}_{BA}^2$ is formed along the direction of travel. It is a wedge-like region and any selectable target that falls within this region is assigned a relatively high probability. With many selectable targets in close proximity from one another, the possibility that the BA model leading to an erroneous prediction is high as noted in [11].

Additionally, as the cursor approaches the true target, the angle $\theta_{i,k}$ can become arbitrarily large leading to small likelihood probabilities and incorrect predictions. Nonetheless, BA tends to make early correct decisions as the users tend to typically head towards the target in the early stages of the pointing task [6, 7].

3.3 Mean Reverting Diffusion (MRD) Model

In a continuous-time, the MAP estimator is based on modeling the cursor movements as a bivariate Ornstein-Uhlenbeck process with a mean-reverting term. It is described by the following stochastic differential equation

$$d\mathbf{c}_t = \boldsymbol{\Lambda}(\boldsymbol{\mu} - \mathbf{c}_t)dt + \boldsymbol{\sigma}_{MRD} d\mathbf{w}_t \tag{7}$$

where $\boldsymbol{\Lambda}$ is a square matrix that sets the mean reversion rate to steer the evolution of the process, $\boldsymbol{\mu}$ is the mean, $\boldsymbol{\sigma}_{MRD}$ is a square matrix that drives the process dispersion and $\mathbf{w}_t$ is a Wiener process [16]. By adopting the above mean reverting diffusion model for the intentionality prediction problem, the mean to which the process should revert to is defined by the location of a selectable target B_i. Hence, $\boldsymbol{\mu} = \mathbf{b}_i$ for the "i^{th}" button and the target, which the cursor is drifting towards, is chosen.

Since the cursor positions are available at discrete times, equations (7) should be discretised. Upon integrating (7) over $\mathcal{T} = [t, t+\tau]$ and then discretising the outcome we obtain

$$\mathbf{c}_{i,k} = e^{-\boldsymbol{\Lambda}\tau_k}\mathbf{c}_{i,k-1} + [\mathbf{I}_2 - e^{-\boldsymbol{\Lambda}\tau_k}]\mathbf{b}_i + \boldsymbol{\varepsilon}_k \tag{8}$$

where $\mathbf{c}_{i,k}$ and $\mathbf{c}_{i,k-1}$ are the state vectors with respect to button B_i at the time instants t_k and t_{k-1} respectively. Whereas, $\tau_k = t_k - t_{k-1}$ is the time step and $\boldsymbol{\varepsilon}_k \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\sigma}^2_{MRD})$ is an additive Gaussian noise. Assuming that the cursor movements along the x and y axis are independent, the $\boldsymbol{\Lambda}$ and $\boldsymbol{\sigma}_{MRD}$ matrices become diagonal, i.e. $\boldsymbol{\Lambda} = diag(\lambda_x, \lambda_y)$ and $\boldsymbol{\sigma}^2_{MRD} = diag(\sigma_x^2, \sigma_y^2)$. It follows that the distribution of the conditional state is given by

$$P(\mathbf{c}_k|\mathbf{c}_{k-1}, B_i) = \mathcal{N}\left(\mathbf{c}_k \middle| \boldsymbol{\Sigma}_{i,k}, \boldsymbol{\Gamma}_k^2\right) \tag{9}$$

such that

$$\boldsymbol{\Sigma}_{i,k} = e^{-\boldsymbol{\Lambda}\tau}\mathbf{c}_{i,k-1} + [\mathbf{I}_2 - e^{-\boldsymbol{\Lambda}\tau_k}]\mathbf{b}_i \tag{10}$$

and

$$\boldsymbol{\Gamma}_k^2 = \left[\frac{1 - e^{-2\boldsymbol{\Lambda}\tau_k}}{2\boldsymbol{\Lambda}}\right]\boldsymbol{\sigma}^2_{MRD}\,. \tag{11}$$

The sought posterior probability is calculated for the MRD model via

$$P(B_i|\mathbf{c}_{k-L:k}) = \frac{P(B_i)\mathcal{N}\left(\mathbf{c}_{L-k}\middle|\boldsymbol{\Sigma}_{i,L-k}, \boldsymbol{\Gamma}_{L-k}^2\right)\prod_{n=k-L+1}^{k}\mathcal{N}\left(\mathbf{c}_n\middle|\boldsymbol{\Sigma}_{i,n}, \boldsymbol{\Gamma}_n^2\right)}{P(\mathbf{c}_{k-L:k})} \tag{12}$$

similar to (6). The reversion rates and the diffusion noise are design parameters that can be tuned to a given data set. It can be noticed that if the pointing cursor is stationary, the MRD MAP reverts to the nearest neighbour model.

3.4 Composite (COM)

The bearing angle model performs poorly when the cursor is moving very slowly since there is no well-defined direction of travel. A composite algorithm uses the BA model whenever the cursor is moving at a velocity that exceeds a certain threshold, i.e. V_T, and switches to the MRD model whenever the cursor speed is below V_T. Selecting the threshold value V_T is a design parameter that requires setting prior to performing the intentionality prediction.

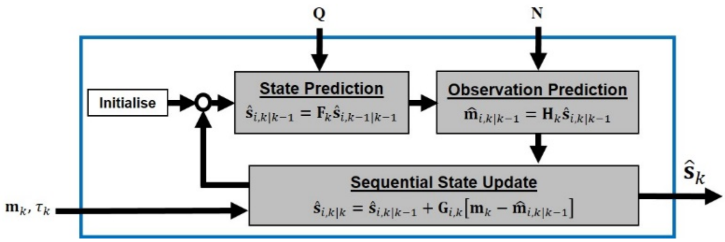


Fig. 1. Simplified block diagram of the LKF; $\mathbf{G}_{i,k}$ is Kalman gain

3.5 Weighted Bearing with Distance (WBD)

This model is motivated by the fact that as the cursor approaches the target the bearing angle can take arbitrary values. On the other hand, the direction of travel tends to be a more reliable indication of the destination when the cursor is relatively far from the intended button [1, 5, 13]. Hence, the weighted bearing with distance model fuses the bearing and distance information where the likelihood probability for two consecutive cursor positions is given by

$$P(\mathbf{c}_k|\mathbf{c}_{k-1}, B_i) = \mathcal{N}\left(\theta_{i,k}\middle|0,\ \kappa_{WBD}^2\Omega_{i,k}^2\right). \tag{13}$$

Similar to the BA model, the divergence of the bearing from the position of B_i is defined by $\theta_{i,k} = \angle(\mathbf{v}_k, \mathbf{b}_i - \mathbf{c}_k)$; $\Omega_{i,k} = 1/\|\mathbf{b}_i - \mathbf{c}_k\|_2$ is the inverse of the Euclidian norm of distance between the cursor's current position and the "i^{th}" button. Hence, if the cursor is in close proximity to a possible target, bigger $\theta_{i,k}$ values can be tolerated due to the resultant $\Omega_{i,k}$ and vice versa. WBD can seamlessly circumvent the unreliable aspects of BA whilst harnessing its ability to predict the correct target in early stages of the pointing process. Similar to (6) and (12), $P(B_i|\mathbf{c}_{k-L:k})$ of WBD can be calculated.

4 Linear Kalman Filter Based Smoothing

Kalman filter is an adaptive Bayesian filtering approach that is widely used for tracking dynamic signals in real-time due to its robustness and low complexity. It is deployed here to remove involuntary cursor movements typically manifested by large deviations from a direct path between the start point and the target location. Such outliners are caused by motor disorders or situational impairments [13]. Based on several studies on pointing tasks in 2-D environments, e.g. [1, 5, 6, 13], it is reasonable to represent the cursor voluntary movements by the nearly constant velocity model. Accordingly, the discretised state dynamics at the time instant t_k are defined by

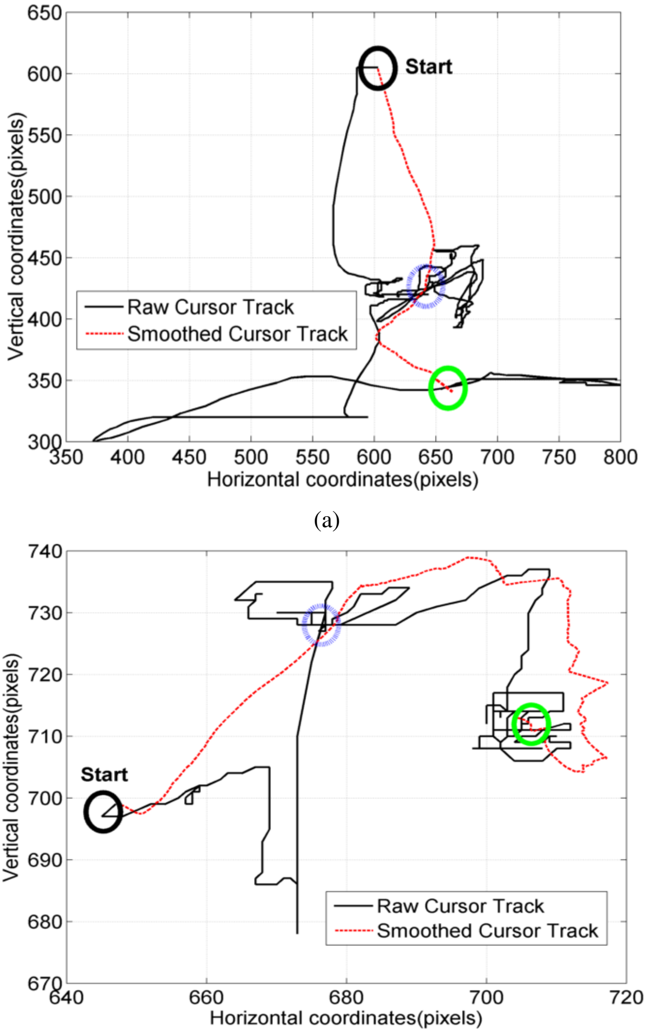
$$\mathbf{s}_k = \mathbf{F}_k \mathbf{s}_{k-1} + \mathbf{e}_k \tag{14}$$

such that $\mathbf{s}_k = \left[x_{t_n} \ \dot{x}_{t_n} \ y_{t_n} \ \dot{y}_{t_n}\right]^T$; $\dot{x}_{t_n}$ and $\dot{y}_{t_n}$ are the velocities along the *x*-*y* axis,

$\mathbf{F}_k = \begin{bmatrix} \hat{\mathbf{F}}_k & \mathbf{0} \\ \mathbf{0} & \hat{\mathbf{F}}_k \end{bmatrix}$, $\hat{\mathbf{F}}_k = \begin{bmatrix} 1 & \tau_k \\ 0 & 1 \end{bmatrix}$, $\mathbf{Q} = \begin{bmatrix} \rho_x \widehat{\mathbf{Q}} & \mathbf{0} \\ \mathbf{0} & \rho_y \widehat{\mathbf{Q}} \end{bmatrix}$, $\widehat{\mathbf{Q}}_x = \begin{bmatrix} \tau_k^3/3 & \tau_k^2/2 \\ \tau_k^2/2 & \tau_k \end{bmatrix}$ and $\mathbf{e}_k$ is a zero mean bivariate additive white Gaussian noise of covariance $\mathbf{Q}$. Design parameters ρ_x and ρ_y set the level of deviations from the constant velocity path; their units is *speed²/time*. The measured cursor positions are modelled as: $\mathbf{m}_k = \mathbf{H}\mathbf{s}_{k-1} + \mathbf{n}_k$ and $\mathbf{H} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$ where $\mathbf{n}_k \sim \mathcal{N}(0, \mathbf{N})$ and $\mathbf{N} = diag\{\sigma_x^2, \sigma_y^2\}$ is the observation noise covariance matrix. Given the Gaussian nature of $\mathbf{s}_k$ and $\mathbf{m}_k$, linear Kalman filter depicted in Fig. 1 is the optimal filter in the minimum mean squared error sense [15]; $\mathbf{Q}$ and $\mathbf{N}$ dictate the level of performed smoothing. The LKF output $\hat{\mathbf{c}}_k = \left[\hat{x}_{t_k} \ \hat{y}_{t_k}\right]^T$, which is the smoothed cursor location at time t_k, is used in the adopted Algorithm 1. Additionally, the resulting smoothed velocity vector $\hat{\mathbf{v}}_k = \left[\hat{v}_x \ \hat{v}_y\right]^T$ can be used for the BA, COM and WBD models. Fig. 2 shows two cursor tracks for a severely motor impaired user with notable tremor attempting two selections on the interface. The effectiveness of the LKF-based-smoothing is clearly demonstrated in the figure.

5 Experiments

The considered prediction models are tested on two data sets pertaining to: 1) user 1 is able bodied and 2) user 2 is severely motor impaired suffering from notable tremor (see the tracks in Fig. 2). They undertook selection tasks similar to the ISO 9241 with multiple distractors on the screen with a typical layout shown in Fig. 3. Users click the button at the centre of the screen Fig. 3a and then the target button appears with other distractors Fig. 3b. The performance of the adopted model is assessed in terms of the percentage of time the predicator makes a correct decision from a partial cursor track, i.e. the accuracy of the prediction. It is noted that at every observation time, e.g. t_k, the predictive model uses the available track (e.g. $\mathbf{c}_{k-L:k}$) and does not assume



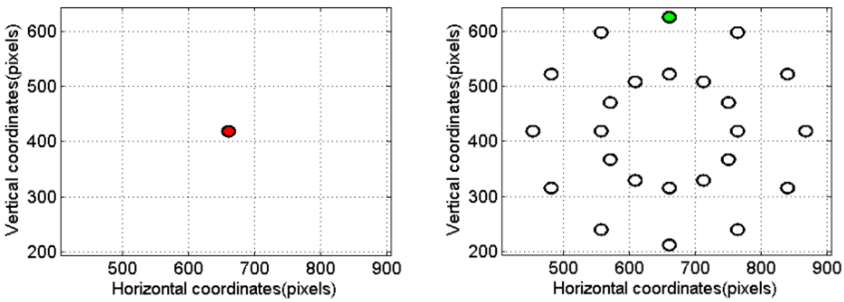
(a)

(b)

Fig. 2. Two raw and smoothed cursor tracks for a severely impaired user. Start point is the black solid circle, target 1 is the dotted blue circle and target 2 is the solid green circle.

knowledge of how much of the entire trajectory, i.e. $\mathbf{c}_{1:M}$, has been completed. Table I exhibits the performance of the considered prediction models with and without the LKF smoother for the two participants. The various design parameters, e.g. L, $\boldsymbol{\Lambda}$, $\boldsymbol{\sigma}_{NN}^2$, $\boldsymbol{\sigma}_{BA}^2$, $\boldsymbol{\sigma}_{MRD}^2$ and κ_{WBD} are obtained from Monte Carlo simulations. This is feasible since each model has a maximum of two parameters that alter its MAP outcome. Design parameters that yield the highest percentage of correct predictions are selected.

It can be noticed from Table 1 that the performance of the examined predictors are drastically affected by the high level of perturbations present in the motor impaired cursor tracks (for example see Fig. 2). Nevertheless, the introduced MRD-based predictor outperforms other methods whereas the WBD and COM models bring notable benefits compared to the conventional NN and BA models. With the adopted prediction algorithms, the system can correctly anticipate the target with over 60% accuracy, i.e. the pointing time can be potentially reduced by 60% for able-bodied users. The achieve gains for the motor impaired user is also significant, particularly after introducing the LKF-based smoother. For example, without the filtering operation all the evaluated predictors perform very poorly for the motor-impaired user given the typical sudden sharp jerks and jumps in the processed trajectories, i.e. neither distant nor heading can give an indication of the intentionality. After introducing LKF, the success rate of the predictors improves remarkably. With the able-bodied user, the LKF has marginal impact on the intentionality prediction results due to the smooth nature of the treated cursor tracks in such cases.



(a) (b)

Fig. 3. An example of an ISO task. Red circle is the start point and green circle is the target.

Table 1. Map Estimator Results for Raw and Smoothed Cursor Pointing Trajectories

	Proportion of Time Correctly Predicting the Target (%)				
Subject	NN	BA	MRD	WBD	COM
Able-bodied	53.1	31.2	61.8	45.13	56.19
Motor impaired	15.3	4.3	18.6	16.4	14.8
Subject	LKF-NN	LKF-BA	LKF-MRD	LKF-WBD	LKF-COM
Able-bodied	53.3	34.7	62.1	47.3	57.9
Motor impaired	38.4	28.3	44.1	38.8	39.7

6 Conclusions

The adopted probabilistic intentionality prediction approach delivers significant reductions in the pointing durations alleviating difficulties experienced by impaired users. This can be particularly beneficial for assistive interfaces by providing visual feedback, e.g. highlighting predicted target(s) or magnifying them, increasing the movement gain, even making a decision on the user's behalf, etc. The simple Kalman filtering approach is shown to effectively smooth highly perturbed cursor trajectories. This study sets a probabilistic framework and serves as an impetus to further research into more advanced Bayesian filtering algorithms that can better stabilise the pointing movements for irregular position measurements and highly non-linear cursor movement models, e.g. sequential Monte Carlo techniques [17]. Additionally, devising more elaborate models that incorporate the target position (similar to MRD) can be highly beneficial since the filtering operation can produce the sought posterior probabilities, i.e. circumvent the need to separate the smoother from the predictor.

References

1. Oirschot, H.K.-V., Houtsma, A.J.M.: Cursor Trajectory Analysis. In: Brewster, S., Murray-Smith, R. (eds.) Haptic HCI 2000. LNCS, vol. 2058, pp. 127–134. Springer, Heidelberg (2001)
2. Bateman, S., Mandryk, R.L., Xiao, R., Gutwin, C.: Analysis and comparison of target assistance techniques for relative ray-cast pointing. International Journal of Human-Computer Studies, 511–532 (2013)
3. Fitts, P.M., Peterson, J.R.: Information capacity of discrete motor responses. J. Exp. Psych. 67, 103–112 (1964)
4. Mackenzie, I.S.: Fitts' law as a research and design tool in human-computer interaction. Journal of Human Computer Interaction 7, 91–139 (1992)
5. Kopper, R., Bowman, D.A., Silva, M.G., McMahan, R.P.: A human motor behavior model for distal pointing tasks. Int. J. of Human Computer Studies 68, 603–615 (2010)
6. Meyer, D.E., Smith, J.E., Kornblum, S., Abrams, R.A., Wright, C.E.: Optimality in human motor performance: ideal control of rapid aimed movements. Psychological Review 8, 340–370 (1988)
7. McGuffin, M.J., Balakrishnan, R.: Fitts's law and expanding targets: Experimental studies and designs for user interfaces. ACM Transactions Computer-Human Interaction 4, 388–422 (2005)
8. Lane, D.M., Peres, S.C., Sándor, A., Napier, A.H.A.: A Process for Anticipating and Executing Icon Selection in Graphical User Interfaces. International Journal of Human Computer Interaction 2, 243–254 (2005)
9. Wobbrock, J.O., Fogarty, J., Liu, S., Kimuro, S., Harada, S.: The Angle Mouse: Target-Agnostic Dynamic Gain Adjustment Based on Angular Deviation. In: Proc. of the 27th International Conference on Human Factors in Computing Systems, New York, pp. 1401–1410 (2009)
10. Murata, A.: Improvement of pointing time by predicting targets in pointing with a PC mouse. International Journal of Human Computer Studies 10, 23–32 (2005)

11. Asano, T., Sharlin, E., Kitamura, Y., Takashima, K., Kishino, F.: Predictive interaction using the Delphian desktop. In: Proc. of the 186th Annual ACM Smp. on User Interface Software and Technology (UIST 2005), New York, pp. 133–141 (2005)
12. Lank, E., Cheng, Y.N., Ruiz, J.: Endpoint prediction using motion kinematicst. In: Proc. of the SIGCHI Conference on Human Factors in Computing System, NY, pp. 637–646 (2007)
13. Keates, S., Hwang, F., Langdon, P., Clarkson, P.J., Robinson, P.: Cursor measures for motion-impaired computer users. In: Proc. of the Fifth International ACM Conference on Assistive Technologies – ASSETS, New York, pp. 135–142 (2002)
14. Ziebart, B., Dey, A., Bagnell, J.A.: Probabilistic pointing target prediction via inverse optimal control. In: Proc. of the ACM Int. Conf. on Intelligent User Interfaces, pp. 1–10 (2012)
15. Haug, A.: Bayesian Estimation and Tracking: A Practical Guide. John Wiley & Sons (2012)
16. Meucci, A.: Review of Statistical Arbitrage, Cointegration, and Multivariate Ornstein-Uhlenbeck. SSRN Preprint 1404905 (2010)
17. Godsill, S.J., Vermaak, J., Ng, W., Li, J.: Models and algorithms for tracking of maneuvering objects using variable rate particle filters. Proc. of IEEE 95, 925–952 (2007)