

# DESTINATION INFERENCE USING BRIDGING DISTRIBUTIONS

Bashar I. Ahmad\*, James Murphy\*, Patrick M. Langdon\*, Robert Hardy\*\*, Simon J. Godsill\*

\* Engineering Department, University of Cambridge, Trumpington Street, Cambridge, UK, CB2 1PZ

\*\* Jaguar Land Rover, Whitley, Coventry, UK

Emails: {bia23, jm362, pml24, sjg30}@cam.ac.uk, rhardy@jaguarlandrover.com

## ABSTRACT

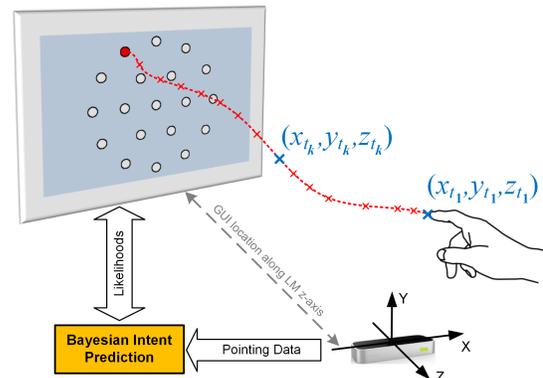
We propose a novel probabilistic inference approach that permits predicting, well in advance, the intended destination of a pointing gesture aimed at selecting an icon on an in-vehicle interactive display. It models the partial 3D pointing track as a Markov bridge terminating at a nominal destination. The solution introduced leads to a low-complexity Kalman-filter-type implementation and is applicable in other areas in which early detection of the destination of a tracked object is beneficial. Data collected in an instrumented vehicle illustrate that the proposed technique can infer the intent notably early in the pointing gesture. This can drastically reduce the pointing task time and visual-cognitive-manual attention required.

**Index Terms**— human computer interactions, intent inference, Kalman filter, bridging distributions.

## 1. INTRODUCTION

Interactive displays such as touchscreens are becoming an integrated part of the modern vehicle environment due to their ability to present large quantities of data associated with In-Vehicle Infotainment Systems (IVIS) [1, 2, 3]. They are also easy to use via instinctive pointing gestures. However, using such displays entails dedicating a considerable amount of attention that would otherwise be available for driving, with serious safety implications [4, 5]. Additionally, due to driving or road conditions the user input can be highly perturbed, leading to erroneous selections, which compromises the system usability and results in further distractions.

In this paper, we propose a Bayesian intent inference approach that allows prediction, early in the pointing gesture, of the intended destination on an in-vehicle interactive display. This can significantly reduce pointing time and effort. Here, the pointing track is modelled as one of several Markov bridges, each incorporating one of the possible destinations, e.g. selectable icons on a GUI displayed on a touchscreen. The path of the pointing finger, albeit random, must end at the intended destination, i.e. it follows a bridge distribution from its start point to the destination. By determining the likelihood



**Fig. 1.** System block diagram with a complete pointing track (dotted line) to select the highlighted GUI icon;  $t_k > t_1$ .

of the observed partial pointing trajectory being drawn from a particular bridge, the probability of each possible destination is evaluated. A gesture tracking sensor, e.g. Leap Motion (LM) [6], is used to produce the 3D track of the pointing finger as depicted in Fig. 1. This system is instrumented in a car to collect data that is used to demonstrate the effectiveness of the inference method. It is noted that in other application areas, such as surveillance and defence, establishing the destination of a tracked object (or the likelihoods of several possible destinations) can be valuable since it constrains the target trajectory and/or offers information on intent or possible threats [7, 8, 9]. Thus, the proposed framework is applicable outside the Human Computer Interactions (HCI) area.

## 2. PROBLEM STATEMENT AND RELATED WORK

Let  $\{\mathcal{D}_i : i = 1, 2, \dots, N\}$  be the set of  $N$  nominal destinations, e.g. GUI icons on an in-vehicle touchscreen. The objective is to determine the probability of each of these items being the intended destination  $I$  of a pointing gesture, given a series of  $k$  measurements,  $m_{1:k} \triangleq \{m_1, m_2, \dots, m_k\}$ , i.e. to calculate  $p(I = i | m_{1:k})$  for each  $i = 1, 2, \dots, N$ . The  $k^{\text{th}}$  observation  $m_k = [\hat{x}_{t_k} \ \hat{y}_{t_k} \ \hat{z}_{t_k}]'$  at time  $t_k$  is the pointing finger 3D coordinates recorded by a gesture-tracker. It is derived from a true, but unknown, underlying finger position  $c_k$ ; its velocity

This work is supported by Jaguar Land Rover (JLR), Whitley, UK.

at time  $t_k$  is notated as  $\hat{c}_k$ . The most probable intended destination at  $t_k$  is given by the Maximum *a Posteriori* estimate

$$i^* = \arg \max_{i=1,2,\dots,N} p(I = i | m_{1:k}). \quad (1)$$

With  $T$  being the total duration of the pointing task, the intent inference at  $t_k$  can reduce the pointing time by  $T - t_k$ .

The benefits of inferring the intended item on a GUI early in a pointing task are widely recognised in HCI, e.g. [10, 11, 12, 13, 14, 15]. Most existing prediction algorithms focus on pointing via a mouse in a 2D set-up. They aim to reduce the pointing time and enable a facilitation strategy such as increasing item size, adjusting activation area, etc. In [16], 2D-based predictors are shown to be unsuitable, or computationally demanding, for 3D pointing. For instance, the linear-regression methods in [13, 14] assume that the destination is always located along the path followed by the pointing object, which is rarely true in pointing gestures [16]. An intent inference approach that models the pointing movement as a Mean Reverting Diffusion (MRD) process was introduced in [16, 17]. The technique proposed in this paper delivers superior prediction results and is robust against changes in the model parameters, unlike the MRD-based method.

Several destination-aware tracking algorithms exist, e.g. [7, 8, 9], where a conventional tracker estimates the target *state* followed by an inference filter to determine its destination. In [9], the monitored spatial area is discretized and a grid of regions is defined. The tracked object can pass through a finite number of the predefined zones. In the 3D pointing tasks considered here, hand movement is free and there are infinite possible paths to the destination, making discretization a burdensome task. Instead, in this paper we introduce a simple low-complexity technique that does not impose trajectories the user's hand or any tracked object ought to follow.

### 3. PROPOSED BRIDGING MODEL

The location of the tracked object, i.e. the pointing fingertip, at the end of the pointing task is that of the intended destination  $\mathcal{D}_I$ . The hidden state of the pointing fingertip at time  $T$  (i.e. at the end of the pointing task) is given by  $s_T = [c'_T \ \dot{c}'_T]'$  where  $c_T$  and  $\dot{c}_T$  are the true finger position and velocity at  $T$  respectively;  $\hat{b}_i = [b'_i \ v'_i]'$  such that  $b_i$  denotes the known location of the  $i^{\text{th}}$  GUI icon in 3D and  $v_i$  is the finger velocity upon contact with the destination. Thus, the probability of  $\mathcal{D}_i$  being the intended destination is

$$\begin{aligned} p(I = i | m_{1:k}) &\propto p(m_{1:k} | I = i)p(I = i) \\ &= p(I = i) \int p(m_{1:k} | s_T = \hat{b}_i)p(T | I = i)dT, \end{aligned} \quad (2)$$

since  $p(m_{1:k} | s_T = \hat{b}_i) = p(m_{1:k} | I = i, T)$ ;  $T$  is unknown.

The priors  $p(I = i)$ ,  $i = 1, 2, \dots, N$ , in (2) are independent of the current trajectory  $m_{1:k}$  and can be learnt from contextual information such as selection history, GUI design, etc.

Henceforth, we assume  $p(I = i) = 1/N$ , although any available priors can easily be used. The objective, then, is to estimate the integral  $\mathcal{A}_i = \int p(m_{1:k} | s_T = \hat{b}_i)p(T | I = i)dT$  for each of the  $N$  possible destinations. A simple quadrature approximation of  $\mathcal{A}_i$  is given by

$$\mathcal{A}_i \approx \sum_n p(m_{1:k} | s_{T_n} = \hat{b}_i)p(T_n | I = i)\Delta T_n \quad (3)$$

where  $\Delta T_n = T_n - T_{n-1}$  and the  $T_n$  are quadrature points, ideally chosen to cover the majority of the probability mass in  $p(T | I = i)$ . More sophisticated quadrature or Monte-Carlo estimates could also be employed. Here, we assume uniformly arrival times priors within a time window determined from the completion time of 57 collected pointing tracks, i.e.  $p(T | I = i) \sim \mathcal{U}(a, b)$ . For simplicity, the finger velocity  $v_i$  at the destination is assumed to be zero for all  $i$ . A more realistic formulation is to integrate over possible values of  $v_i$ , but this is not considered here.

#### 3.1. Motion and Observation Models

The state of the user's finger  $s_k = [c'_k \ \dot{c}'_k]'$  at time  $t_k$  is assumed to follow the linear Gaussian motion model

$$s_k = F_k s_{k-1} + \varepsilon_k \quad (4)$$

with  $\varepsilon_k \sim \mathcal{N}(0, Q_k)$ . This general form permits many useful motion models, the simplest of which is the (near) constant velocity model, which is the solution of the continuous-time stochastic differential equation

$$ds_t = \begin{bmatrix} 0_3 & I_3 \\ 0_3 & 0_3 \end{bmatrix} s_t dt + \begin{bmatrix} 0_3^v \\ \sigma \end{bmatrix} dW_t$$

where  $dW_t$  is the instantaneous change of a standard Brownian motion at time  $t$ ,  $0_3$  is a  $3 \times 3$  zero matrix,  $I_3$  is a  $3 \times 3$  identity matrix and  $0_3^v$  is a  $3 \times 1$  zero vector. The corresponding  $F_k$  and  $Q_k$  matrices in equation (4) are given by  $F_k = M(\Delta_k)$  and  $Q_k = R(\Delta_k)$ , where the time step  $\Delta_k = t_k - t_{k-1}$  (which can vary, allowing asynchronous observations), and

$$M(p) = \begin{bmatrix} I_3 & pI_3 \\ 0_3 & I_3 \end{bmatrix}, \quad R(p) = \sigma^2 \begin{bmatrix} \frac{1}{3}p^3I_3 & \frac{1}{2}p^2I_3 \\ \frac{1}{2}p^2I_3 & pI_3 \end{bmatrix}, \quad (5)$$

with  $\sigma$  setting the motion model state transition noise level. The movements in the  $x$ ,  $y$  and  $z$  dimensions are considered to be independent from one another.

Observations are assumed to be a linear function of the current system state with additive Gaussian noise, such that

$$m_k = H_k s_k + \eta_k \quad (6)$$

with  $\eta_k \sim \mathcal{N}(0, V_k)$ . For the LM pointing fingertip data, we have  $H_k = [I_3 \ 0_3]$  for all  $k$ , since LM sensor makes direct (noisy) observations of the true finger position  $c_k$ . It is noted that other motion models suitable for intent inference that could be utilised in this framework. Those include the destination-reverting models in [16] and the linear portion of the perturbation removal model in [18].

### 3.2. Hidden State and Likelihood Evaluation

Without conditioning information, the distribution of the hidden state  $s_k$  given observations  $m_{1:k}$  in equations (4) and (6) can be calculated by a standard Kalman Filter (KF) as per

$$p(s_k | m_{1:k}) = \mathcal{N}(s_k; \mu_{k|k}^{\text{KF}}, \Sigma_{k|k}^{\text{KF}}),$$

with (using the ‘correct’ step of the Kalman filter),

$$\mu_{k|k}^{\text{KF}} = \mu_{k|k-1}^{\text{KF}} + K_k(m_k - H_k \mu_{k|k-1}^{\text{KF}}) \quad (7)$$

$$\Sigma_{k|k}^{\text{KF}} = (\mathbf{I}_6 - K_k H_k) \Sigma_{k|k-1}^{\text{KF}} \quad (8)$$

$$K_k = \Sigma_{k|k-1} H_k' \left( H_k \Sigma_{k|k-1}^{\text{KF}} H_k' + V_k \right)^{-1}.$$

Here,  $\mu_{k|k-1}$  and  $\Sigma_{k|k-1}$  are derived from the inferred system distribution at  $t-1$ , given by the prediction step of the KF:

$$\mu_{k|k-1}^{\text{KF}} = F_k \mu_{k-1|k-1}^{\text{KF}} \quad (9)$$

$$\Sigma_{k|k-1}^{\text{KF}} = F_k \Sigma_{k-1|k-1}^{\text{KF}} F_k' + Q_k. \quad (10)$$

When  $k=1$ , these quantities are given by the priors, so that  $\mu_{1|0}^{\text{KF}} = \mu_{\text{prior}}$  and  $\Sigma_{1|0}^{\text{KF}} = \Sigma_{\text{prior}}$ . They represent prior knowledge of track start position, i.e.  $p(s_1) \sim \mathcal{N}(\mu_{\text{prior}}, \Sigma_{\text{prior}})$ .

In order to condition on the system state at the destination arrival time,  $s_T$ , it is necessary to evaluate the density  $p(s_T | s_k)$  for the current tracked object state (and arrival time). For motion models derived from continuous-time processes, such as the near constant velocity model used here, this is possible by direct integration of the motion model (which is possible in the linear time-invariant Gaussian case). For the near constant velocity model, this is given by

$$p(s_T | s_k) = \mathcal{N}(s_T; M_k s_k, R_k),$$

where  $M_k = M(T-t_k)$  and  $R_k = R(T-t_k)$  from equation (5), and  $T-t_k$  is the time step between the  $T^{\text{th}}$  and  $t_k^{\text{th}}$  observations. Alternatively, forward or backward recursions can be formed in terms of  $F_{2:T}$ , and  $Q_{2:T}$ , which can be used with discrete models without a continuous-time interpretation.

Subsequently, the conditional *predictive* distribution of  $s_k$  given the  $k-1$  observations and the intended destination (which specifies  $s_T$ ) can be shown to reduce to

$$p(s_k | m_{1:k-1}, s_T) = \mathcal{N}(s_k; \mu_k^*, \Sigma_k^*) \quad (11)$$

$$\mu_k^* = \mu_{k|k-1}^{\text{KF}} + K_k^*(s_T - M_k \mu_{k|k-1}^{\text{KF}}), \quad (12)$$

$$\Sigma_k^* = (\mathbf{I}_6 - K_k^* M_k) \Sigma_{k|k-1}^{\text{KF}}. \quad (13)$$

$$K_k^* = \Sigma_{k|k-1} M_k' \left( M_k \Sigma_{k|k-1} M_k' + R_k \right)^{-1}.$$

This can be seen by analogy to the ‘correct’ step of the standard Kalman filter [19].

By taking the latest observation into account, the *correction* stage (taking account of  $m_k$ ) can be shown to be

$$p(s_k | m_{1:k}, s_T) = \mathcal{N}(s_k; \mu_k, \Sigma_k) \quad (14)$$

where  $\mu_k = \mu_k^* + \tilde{K}_k(y_k - H_k \mu_k^*)$ ,  $\Sigma_k = (\mathbf{I}_6 - \tilde{K}_k H_k) \Sigma_k^*$  and  $\tilde{K}_k = \Sigma_k^* H_k' (H_k \Sigma_k^* H_k' + V_k)^{-1}$ . This can also be seen by analogy with the ‘correct’ step of the Kalman filter [19] noting that  $p(s_k | m_{1:k}, s_T) \propto \mathcal{N}(m_k; H_k s_k, V_k) \mathcal{N}(s_k; \mu_k^*, \Sigma_k^*)$ .

Together with the standard KF, the above predict and correct steps allow the conditional distribution of finger position to be calculated at the time of each observation, conditional on the destination and arrival time. It remains to calculate

$$p(m_{1:k} | I = i, T) = \prod_{l=1}^k p(m_l | m_{1:l-1}, s_T) \quad (15)$$

where it can be shown that

$$\begin{aligned} p(m_k | m_{1:k-1}, s_T) &= \int p(m_k | s_k) p(s_k | m_{1:k-1}, s_T) ds_k \\ &= \mathcal{N}(m_k; H_k \mu_k^*, H_k \Sigma_k^* H_k'). \end{aligned} \quad (16)$$

This is equivalent to the prediction error decomposition in the KF [19]. Note that if likelihood calculation is the objective of filtering, the corrective step in equation (14) is not required.

Using the likelihood in equation (15), the probability of each nominal destination can be evaluated via equations (2) and (3) upon arrival of a new observation. Algorithm 1 gives a sequential implementation of the proposed method.

---

#### Algorithm 1 Sequential Intent Inference

---

```

Set  $L_0^{i,n} = 1$  for all  $i$  (targets),  $T_n$  (end times)
for (each observation)  $k = 1, \dots, k_{\text{max}}$  do
  for (each possible destination)  $i = 1 \dots N$  do
    for (each end time)  $T_n$  with  $n = 1, \dots, n_{\text{max}} - 1$  do
      (In the following calculations,  $s_{T_n} = [b_i' v_i']'$ )
      - Calculate  $\mu_{k|k-1}^{i,n}, \Sigma_{k|k-1}^{i,n}$  via equations (9), (10)
      - Calculate and store  $\mu_{k|k}^{i,n}, \Sigma_{k|k}^{i,n}$  via equations (7) and (8)
      - Calculate  $\mu_k^{*,i,n}, \Sigma_k^{*,i,n}$  via equations (12), (13)
      - Calculate likelihood using equation (15):
         $L_k^{i,n} = L_{k-1}^{i,n} \mathcal{N}(m_k; H_k \mu_k^{*,i,n}, H_k \Sigma_k^{*,i,n} H_k')$ 
    end for
    Approximate integration over  $T$  via equation (3):
     $L_k^i = \sum_{n=1}^{n_{\text{max}}-1} L_k^{i,n} p(T = T_n | I = i) (T_n - T_{n-1})$ 
    Let  $\tilde{P}_k^i = L_k^i \times p(I = i)$ 
  end for
  Calculate target probabilities:  $p(I = i | m_{1:k}) \approx \tilde{P}_k^i / \sum_i \tilde{P}_k^i$ 
end for

```

---

## 4. RESULTS

Here, we assess the performance of the proposed Bridging Distributions (BD) predictor for 57 pointing tracks collected in an instrumented car driven over various road types. The data pertains to four passengers undertaking pointing tasks to select highlighted GUI icons displayed on the in-vehicle touchscreen. The layout of the GUI is similar to that in Fig. 1 with 21 selectable circular icons that are less than 2 cm apart.

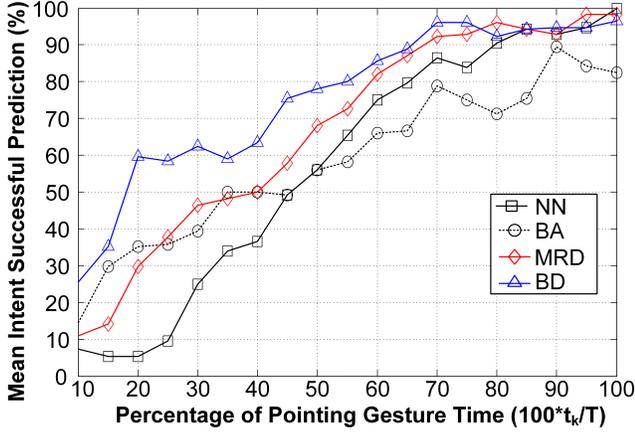


Fig. 2. Mean percentage of destination successful prediction.

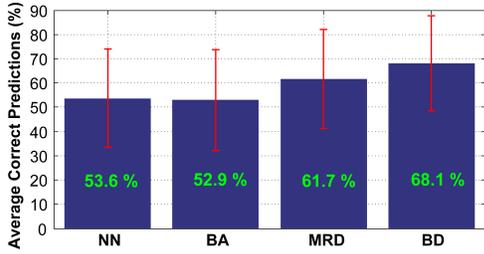


Fig. 3. Gesture portion (in time) with successful prediction.

The predictor performance is evaluated in terms of its ability to successfully establish the intended icon  $I$  via the MAP estimator in (2), i.e. how early in the pointing gesture the predictor assigns the highest probability to the intended GUI icon  $I$ . This is depicted in Fig. 2 against the percentage of completed pointing gesture (in time) and averaged over all pointing tasks considered. Fig. 3 shows the proportion of the total pointing gesture (in time) for which the predictors correctly established the intended destination. To represent the level of average prediction uncertainty, Fig. 4 displays the mean of the uncertainty metric given by  $\vartheta(t_k) = -\log_{10} p(I = i | m_{1:k})$  where  $i$  is the true intended destination; it is expected that  $\vartheta(t_k) \rightarrow 0$  as  $t_k \rightarrow T$  for a reliable predictor.

In addition to the MRD model in [16], the Nearest Neighbour (NN) and Bearing Angle (BA) benchmark methods are also examined. In the former, the  $\mathcal{D}_i$  closest to the pointing finger position is assigned the highest probability and vice versa; i.e.  $p(m_k | I = i) = \mathcal{N}(m_k; b_i, \sigma_{NN}^2)$  where  $\sigma_{NN}^2$  is the covariance of the multivariate normal distribution. In BA,  $P(m_k | m_{k-1}, I = i) = \mathcal{N}(\theta_k; 0, \sigma_{BA}^2)$  where  $\theta_k = \angle(m_k - m_{k-1}, b_i)$  is the angle to target and  $\sigma_{BA}^2$  is a design parameter. It assumes that the cumulative angle to the intended destination should be minimal. To ensure fair comparisons, design parameters that produce the best prediction performance for the considered models are applied.

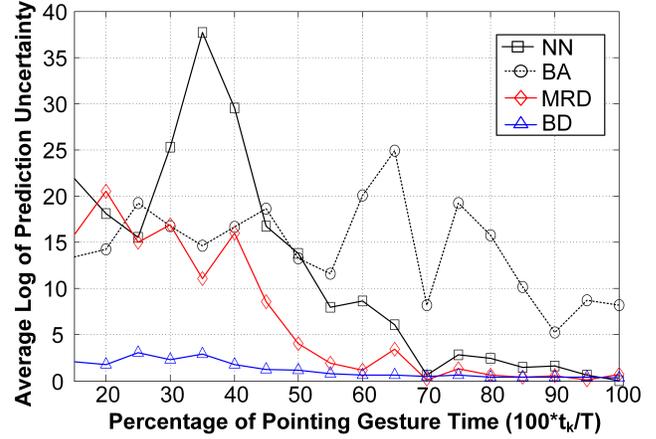


Fig. 4. Average log prediction uncertainty.

Fig. 2 shows that the introduced bridging-distributions-based inference achieves the earliest successful intent predictions. This is particularly visible in the first 75% of the pointing gesture where notable reductions in the pointing time can be achieved and pointing facilitation regimes can be most effective. The performance gap between the various predictors diminishes towards the end of the pointing task. An exception is the BA model where the reliability of the heading angle as a measure of intent declines as the the pointing finger gets closer to the target [16]. Fig. 3 shows that the BD approach delivers the highest overall correct predictions across the pointing trajectories (NN and BA performances are similar over the relatively large data set considered).

Fig. 5 illustrates that the proposed BD model makes correct predictions with significantly higher confidence throughout the pointing task, compared to other methods. Overall, Figs. 2, 3 and 4 demonstrate that the BD inference approach introduced predicts, well in advance, the intent of an in-vehicle pointing gesture, e.g. only 20% into the gesture in 60% of cases, which can reduce pointing time/effort by 80%.

## 5. CONCLUSION

This paper introduces a novel framework for low-complexity, reliable intent inference. The early prediction of tracked object (pointing finger) destination, can notably reduce the pointing time and attention required to interact with in-vehicle displays. As display interaction becomes increasingly prevalent in modern vehicles, small improvements in pointing task efficiency such as reducing the pointing time by a few milliseconds via improved prediction quality, will have substantial aggregate benefits and enhance safety, especially for a driving user. This study serves as an impetus to further research and calls for a full experimental study to identify pointing facilitation techniques that best leverage the prediction results and quantify the benefits on the overall user experience/safety.

## 6. REFERENCES

- [1] Catherine Harvey and Neville A Stanton, *Usability evaluation for in-vehicle systems*, CRC Press, 2013.
- [2] Fong-Gong Wu, Hsuan Lin, and Manlai You, “Direct-touch vs. mouse input for navigation modes of the web map,” *Displays*, vol. 32, no. 5, pp. 261–267, 2011.
- [3] Gary E Burnett and J Mark Porter, “Ubiquitous computing within cars: designing controls for non-visual use,” *International Journal of Human-Computer Studies*, vol. 55, no. 4, pp. 521–531, 2001.
- [4] Sheila G Klauer, Thomas A Dingus, Vicki L Neale, Jeremy D Sudweeks, and David J Ramsey, “The impact of driver inattention on near-crash/crash risk: An analysis using the 100-car naturalistic driving study data,” National Highway Traffic Safety Administration, DOT HS 810 5942006, 2006.
- [5] Yulan Liang and John D Lee, “Combining cognitive and visual distraction: Less than the sum of its parts,” *Accident Analysis & Prevention*, vol. 42, no. 3, pp. 881–890, 2010.
- [6] Lee Garber, “Gestural technology: Moving interfaces in a new direction [technology news],” *Computer, IEEE*, vol. 46, no. 10, pp. 22–25, 2013.
- [7] David A Castanon, Bernard C Levy, and Alan S Willsky, “Algorithms for the incorporation of predictive information in surveillance theory,” *International Journal of Systems Science*, vol. 16, no. 3, pp. 367–382, 1985.
- [8] Claudio Piciarelli, Christian Micheloni, and Gian Luca Foresti, “Trajectory-based anomalous event detection,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 18, no. 11, pp. 1544–1554, 2008.
- [9] Mustafa Fanaswala and Vikram Krishnamurthy, “Detection of anomalous trajectory patterns in target tracking via stochastic context-free grammars and reciprocal process models,” *IEEE Journal of Selected Topics in Signal Processing*, vol. 7, no. 1, pp. 76–90, 2013.
- [10] Atsuo Murata, “Improvement of pointing time by predicting targets in pointing with a PC mouse,” *IJHCI*, vol. 10, no. 1, pp. 23–32, 1998.
- [11] Michael J McGuffin and Ravin Balakrishnan, “Fitts’ law and expanding targets: Experimental studies and designs for user interfaces,” *ACM Transactions on Computer-Human Interaction*, vol. 12, no. 4, pp. 388–422, 2005.
- [12] David Lane, S Peres, Anikó Sándor, and H Napier, “A process for anticipating and executing icon selection in graphical user interfaces,” *IJHCI*, vol. 19, no. 2, pp. 241–252, 2005.
- [13] Takeshi Asano, Ehud Sharlin, Yoshifumi Kitamura, Kazuki Takashima, and Fumio Kishino, “Predictive interaction using the delphian desktop,” in *Proc. of the ACM Symp. on User Interface Software and Technology*. ACM, 2005, pp. 133–141.
- [14] Edward Lank, Yi-Chun Nikko Cheng, and Jaime Ruiz, “Endpoint prediction using motion kinematics,” in *Proc. of the SIGCHI Conf. on Human factors in Computing Systems*, 2007, pp. 637–646.
- [15] Brian Ziebart, Anind Dey, and J Andrew Bagnell, “Probabilistic pointing target prediction via inverse optimal control,” in *Proc. of the 2012 ACM Int. Conf. on Intelligent User Interfaces*, 2012, pp. 1–10.
- [16] Bashar I. Ahmad, James K. Murphy, Patrick M. Langdon, and Simon . Godsill, “Bayesian target prediction from partial finger tracks: Aiding interactive displays in vehicles,” in *15th International Conference on Information Fusion (FUSION ’14)*. IEEE, 2014, pp. 1–7.
- [17] Bashar I. Ahmad, Patrick M. Langdon, Simon J. Godsill, Robert Hardy, Eduardo Dias, and Lee Skrypchuk, “Interactive displays in vehicles: Improving usability with a pointing gesture tracker and Bayesian intent predictors,” in *Proc. of International Conference on Automotive User Interfaces and Interactive Vehicular Applications (AutomotiveUI ’14)*. ACM, 2014, pp. 1–8.
- [18] Bashar I. Ahmad, James K. Murphy, Patrick M. Langdon, and Simon J. Godsill, “Filtering perturbed in-vehicle pointing gesture trajectories: Improving the reliability of intent inference,” in *Proc. of IEEE International Workshop on Machine Learning for Signal Processing (MLSP ’14)*. IEEE, 2014, pp. 1–6.
- [19] Anton J Haug, *Bayesian Estimation and Tracking: A Practical Guide*, John Wiley & Sons, 2012.