Int. J. Mach. Learn. & Cyber.
DOI 10.1007/s13042-015-0488-5

ORIGINAL ARTICLE

Position stabilisation and lag reduction with Gaussian processes in sensor fusion system for user performance improvement

Shimin Feng[1] · **Roderick Murray-Smith**[1] · **Andrew Ramsay**[1]

Received: 11 May 2015 / Accepted: 23 December 2015

Abstract In this paper we present a novel Gaussian Process (GP) prior model-based sensor fusion approach to dealing with position uncertainty and lag in a system composed of an external position sensing device (Kinect) and inertial sensors embedded in a mobile device for user performance improvement. To test the approach, we conducted two experiments: (1) GPs sensor fusion simulation. Experimental results show that the novel GP sensor fusion helps improve the accuracy of position estimation, and reduce the lag (0.11 s). (2) User study on a trajectory-based target acquisition task in a spatially aware display application. We implemented the real-time sensor fusion system by augmenting the Kinect with a Nokia N9. In the trajectory-based interaction experiment, each user performed target selection tasks following a trajectory in (a) the Kinect system and (b) the sensor fusion system. In comparison with the Kinect time-delay system, our system enables the user to perform the task easier and faster. The MSE of target selection was reduced by 38.3 % and the average task completion time was reduced by 26.7 %.

Keywords Gaussian processes · Human-computer interaction · Sensor fusion · Uncertainty · User interfaces

✉ Shimin Feng
shiminf@dcs.gla.ac.uk

Roderick Murray-Smith
Roderick.Murray-Smith@glasgow.ac.uk

Andrew Ramsay
Andrew.Ramsay@glasgow.ac.uk

[1] School of Computing Science, University of Glasgow, Glasgow G12 8QQ, UK

1 Introduction

The Microsoft Kinect sensor can be enhanced with the built-in inertial sensors in a mobile device [3, 7, 8]. We will explore the complementary properties of these sensors and apply a Gaussian Process prior model for fusing the low-sampling-rate position sensed by the Kinect and the higher frequency accelerations measured by the mobile inertial sensors. The sensor fusion helps stabilise the skeleton joint position and reduce the lag.

As the advanced sensors are becoming ubiquitous, many human-computer interaction systems are composed of a range of elements which observe the world via a diverse set of sensors [41]. These sensors might work at a range of sampling rates, depending on power constraints, they may measure different derivatives of measurands (e.g. position, velocity, acceleration) in the world and they might have different noise characteristics [17]. If we can fuse information from such systems in an efficient and principled manner, we can potentially improve the capability of the system without adding extra sensing hardware. A concrete example of this is integration of inertial data from mobile devices such as phones or tablets with position sensing from an embedded Microsoft Kinect sensor, but the same principle can be found in many systems [47].

The Microsoft Kinect can be used for skeleton tracking and the research is useful for novel styles of interaction [59]. However, the two problems with the Microsoft Kinect skeleton tracking include the joint position uncertainty and the latency (0.1 s) [1]. For human motion tracking with the Kinect, the noisy position measurement is a common problem [4]. Advanced sensor fusion techniques could improve the usability by providing more accurate position data, but external states cannot be known with absolute

Published online: 11 January 2016

accuracy and uncertainty always persists [56]. Besides sensor sources, hand tremor and human motor variability will also affect the sensor measurements and induce uncertainty [46].

To address this problem, we need to apply filtering or sensor fusion techniques. However, filtering will introduce lags, which reduces the system responsiveness [4], potentially causing lower satisfaction and poor productivity among users [42]. For instance, in virtual reality, high latency can induce unpleasant user experience [5]. Besides, to minimize both jitter and lag with a filter in the Kinect system is challenging. However, with additional, complementary sensors, e.g. the inertial sensors [27], we can improve the position estimation, reducing the jitter and the lag of the system.

In order to fuse the Kinect sensor and the inertial sensors for state estimation, we need dynamical system modelling techniques. Bayesian filtering is a general framework for recursively estimating the state of a dynamic system [21]. The basic idea of Bayes filtering is that we estimate the state of the system with probabilistic models including the state transition model and the observation model. For instance, the Kalman filter and its variants (EKF and UKF) have been widely used for filtering and sensor fusion [54, 55, 60].

Although Bayesian parametric filters, e.g. the Kalman filter, are efficient, the data flexibility and the predictive capabilities are limited [22]. In recent years, Bayesian nonparametric models have become popular. Gaussian Process (GP) priors are examples of nonparametric models and have been applied for regression problems such as robotics and human motion analysis [21, 52].

One of the drawbacks of applying Gaussian processes for dynamical system modelling is that it is computationally expensive. The major computation in a GP is the inversion of the covariance matrix. Our model is an autoregressive model and the covariance matrix is a fixed matrix for the constant sampling rate (90 Hz), making it very computationally efficient.

In this work, our primary contribution is to propose a GP prior model-based sensor fusion approach to dealing with the position uncertainty and lag problem in a conventional position sensing system (Kinect). We propose a variation of a Gaussian Process prior model [38] that incorporates the low-sampling-rate measurements and the high-sampling-rate derivatives in multi-rate sensor fusion. It takes into account the different sampling rates and the different noise characteristics of the Kinect sensor and the inertial sensors. Based on the GP model, the system can infer the position (and its uncertainty) more accurately and with less delay than other filters. To test this, we built an experimental setup where users followed trajectories and performed target selection in a spatially aware display application. The targeting action of the user was facilitated with the sensor fusion prediction. Experimental results show that the improved accuracy, and reduced delay from the sensor fusion system, compared to the filtered system means that users can acquire the target more rapidly, and with fewer errors. They also reported improved performance in subjective questions.

2 Related work

We consider the problem of fusing the Kinect sensor and the built-in inertial sensors in a mobile device for improving the state estimation in a non-linear dynamical system and demonstrate the benefits of the GP prior model-based sensor fusion in a spatially aware display application. We cover related work including multisensor data fusion and probabilistic approaches, and other related work including mobile spatial interaction and spatially aware displays, and target acquisition.

This work focuses on sensor fusion with GPs instead of optimizing and improving the surrogate modelling [10, 11, 15] to improve the GP regression results. The surrogate modelling has been investigated in literature. Forrester et al. [11] investigated the applications of correlated Gaussian process based approximations to optimization and demonstrated that correlating analyses at multiple levels of fidelity can improve surrogate modelling. The use of surrogate models in engineering design was presented in [10]. The surrogate modelling was also investigated in [15] that used gradient-enhanced kriging and a generalized hybrid bridge function to improve the variable-fidelity surrogate modelling. In this paper, we used the standard optimization algorithm to estimate the hyperparameters of the GP, proposed and generalized the GP prior model-based approach to modelling the sensor fusion system. Cokriging methods have been investigated to take advantage of the covariance between related regionalized variables [13]. GPDM [52] and the proposed GP prior model both deal with human motion modelling. However, they have different focuses. Wang et al. [52] proposed GPDM to learn models of human pose and motion from high-dimensional motion capture data. Instead of learning a representation of the nonlinear dynamics in human motion, we proposed the GP model to fuse data from different sensors and to improve user performance with the GP-based sensor fusion approach. As the sensor measurements are noisy, we apply the GPs to fuse data, taking account of the complementary properties of the sensors and the smoothness of human motion measured by multiple sensors.

2.1 Sensor fusion

Multisensor data fusion combines data from multiple sensors, and related information from associated databases, to achieve improved accuracies and more specific inferences than could be achieved by the use of a single sensor alone [14]. The sensor data can be combined at the data level, the feature level and the decision level [14, 23]. It requires interdisciplinary knowledge and techniques drawn from digital signal processing, statistical estimation and probability, control theory and artificial intelligence [14, 28]. It has widespread applications including military applications, e.g. multitarget tracking [43], and civilian applications, e.g. robotics [48]

The role of sensor fusion is to minimize the user's uncertainty of information [26, 32]. For any location-aware system, position uncertainty is critical to the effective use and acceptance of the system [2, 46]. In robotics, a primary challenge is to deal with uncertainty, which arises for many reasons, including the limitations of the model, the limited perceptual capabilities of the sensors and the noisy measurements, and the approximate nature of the algorithm. Probabilistic approaches, among which Kalman filter is a popular method are described in [48].

Sensor fusion, combining position sensor and inertial sensors has been applied in inertial navigation system (INS) and the motion control of robots [19]. For inertial navigation applications, an INS-GPS integration system combines INS measurements with GPS, providing greater precision than any single system alone [49]. For motion control of robots, the combination of vision sensors and inertial sensors has been investigated in literature [6, 18]. Integration of visual and inertial sensing modalities opens new application directions for robotics and other fields [6].

Probabilistic data fusion methods, e.g. the Kalman filter and its variants, the Monte Carlo and the Sequential Monte Carlo, are widely used in robotics. Although many sensor fusion algorithms exist in literature, there is no standard and well-established evaluation framework to assess the performance of data fusion algorithms [20].

The Gaussian Process prior has been studied in [29]. The Kalman filter can be seen as a special case of Gaussian processes (GPs) [24, 40]. However, the Kalman filter uses the physical state equations, that is, it uses the state transition model and the measurement model for prediction and updating respectively while the covariance function in the GP defines similarity between data-points, allowing us to make predictions based on the closeness of these data-points.

Gaussian processes have been widely used for sensor fusion. In [44], Gaussian processes provide an approach to nonparametric modelling which allows a straightforward combination of function and derivative observations in an empirical model. In [33], the transformed Gaussian Process priors were applied for estimating the derivatives of noisy sensor measurements and sensor fusion. In [51], Gaussian processes were applied for terrain data fusion.

2.2 Other related work

2.2.1 Mobile spatial interaction and spatial aware display

Ubiquitous computing provides the potential to associate information with physical spaces. Mobile spatial interaction is an emerging field in the location-aware applications [46]. Spatially aware displays provide access to more information by mapping physical movement of the device to the movement in virtual space. In this way, the screen of handheld device is like a window, through which the user can see the virtual information stored in the physical space. Fitzmaurice proposed this idea in 1993 [9]. Peephole displays [58] show a movable window on the large 2D virtual space and augment the physical space around a user with digital information.

2.2.2 Target acquisition

Target acquisition has been studied in HCI and plays an important role in mobile augmented reality (AR) applications [39]. However, lags significantly degrade human performance in target acquisition tasks [30, 53]. Besides, position uncertainty, i.e. spatial jitter, may also affect performance [37]. Latency and jitter adversely affect human performance in 2D pointing tasks with stationary targets [36].

3 Gaussian Process model for sensor fusion

3.1 GP regression

3.1.1 GP prior prediction

Consider a nonlinear dynamical system $g(x)$ with known inputs x and observed outputs y. At each time instant i, the measurement y_i is a function of the latent state x_i.

$$y_i = g(x_i) + \varepsilon_i, \tag{1}$$

where ε_i denotes Gaussian system noise, and $\varepsilon_i \sim \mathcal{N}(0, \sigma^2)$, where σ is the standard deviation.

Given a set of N training data-points $\{x_i, y_i, i = 1, \ldots, N\}$, where $X = [x_1, \ldots, x_N]^T$ is an N-dimensional vector of inputs. In this work, the time instants are used as the training inputs. $Y = [y_1, \ldots, y_N]^T$ is a vector of output data

and assumed to be drawn from an N-dimensional normal distribution,

$$Y \sim \mathcal{N}(0, \Sigma), \tag{2}$$

where Σ is the $N \times N$ covariance matrix, the elements of which are functions of inputs X. The covariance function is of the form

$$\operatorname{cov}(f(x_i), f(x_j)) = v_0 \exp\left(-\sum_k \omega_k \left(x_{i,k} - x_{j,k}\right)^2\right) + \sigma_n^2 \delta_{ij}, \tag{3}$$

where $\{v_0, \omega_k, \sigma_n^2\}$ are the hyperparameters. v_0 represents the signal variance. $k = 1$. ω_1 is related with the length scale and σ_n^2 represents the noise variance.

Based on the training input X, the covariance matrix C can be determined according to (3). Given a new input vector x^*, we can find the predictive distribution of the corresponding output y^* according to (4) and (5).

$$\mu(x^*) = C(x^*, X)\left[C(X, X) + \sigma_n^2 I\right]^{-1} Y, \tag{4}$$

where σ_n^2 represents the variance of the Gaussian noise defined in (1).

$$\sigma^2(x^*) = C(x^*, x^*) - C(x^*, X)\left[C(X, X) + \sigma_n^2 I\right]^{-1} C(X, x^*), \tag{5}$$

where $C(x^*, x^*)$ represents the covariance matrix between the test inputs and themselves. $C(x^*, X)$ represents the covariance matrix between the test inputs and the training inputs. $C(X, X)$ represents the covariance matrix between the training inputs and themselves.

3.1.2 Transformations of Gaussian Process priors

Instead of observing Y directly, we assume that the observation m is a transformation of the latent variables y. In the continuous case,

$$output = \int_\Omega system \times input \, \mathrm{d}\Omega, \tag{6}$$

$$m(t) = \int K(t, x) y(x) \, \mathrm{dx}, \tag{7}$$

which in discrete sampled form is

$$m_k = \sum_{i=1}^{N} K_{ki} Y_i. \tag{8}$$

The input-output relationship of a continuous system is expressed in (6), where the input is convolved with the system to yield the output and Ω is defined as the independent variable (the domain). In (7), we define a kernel function $K(t, x)$. The sensor characteristics described in $K(t, x)$ could be nonlinear, changing with state x, while retaining a linear transformation on discretisation. Note that although the discretised form K is a linear transformation, the original kernel $K(t, x)$ could represent a nonlinear mapping. Equation (7) is defined as a general form to represent the relationship between the transformation m and the latent variables y. Its discrete sampled form is (8). In other words, for the vector of latents Y, we observe outputs $M = KY$ with known K, and Y being the unknown state of the latent GP. For instance, this could correspond to an inverse problem such as image restoration, where the observable is the image, the system is the lens, and the scenery is the input. The K represents the operations, e.g. filters, or differentiation, applied to the latent variables before observation, reflecting sensor characteristics or intervening transformation of the states.

The vector M is drawn from an n-dimensional normal distribution:

$$M \sim \mathcal{N}\left(0, K\Sigma K^T + \Sigma_M\right), \tag{9}$$

where Σ is the covariance matrix defined in (2) and Σ_M is the diagonal matrix of observation variances.

The transformed GP priors approach can be generalized to solve the data fusion problem in a wider range of sensor fusion systems. Although the transformations are limited to approximations of derivative transformations in this paper, this method can be generalized through the transformation matrix K. In this paper, we have two sources, that is, the positioning sensor and the mobile device that measures the acceleration. In the case of observation M composed of a number of vectors $M_i = K_i Y$, we can generalize (8) in the following way.

$$\begin{bmatrix} M_1 \\ M_2 \\ \vdots \\ M_n \end{bmatrix} = \begin{bmatrix} K_1 \\ K_2 \\ \vdots \\ K_n \end{bmatrix} Y, \tag{10}$$

where M_i represents the measurements sensed by sensor i, $i = 1, \ldots, n$. K_i denotes the corresponding transformation matrix. In this way, we can apply the transformations of GP priors to fuse data from multiple sources.

3.2 Gaussian Process model for multi-rate sensor fusion

3.2.1 Problem statement for dynamical system modelling

We consider the situation when the user holds a mobile device in the hand and tries to explore the digital information embedded in the Kinect space in the room. The system state desired to estimate is the position of the hand (phone). The problem is that the Kinect position measurements are noisy and delayed. We aim to increase the stability of the position and reduce the lag by using the GP

prior model-based sensor fusion approach to fusing the low-sampling-rate position sensed by the Kinect and the higher frequency acceleration measured by mobile inertial sensors. We define the Kinect latency to be 0.1 s [1, 25].

The human and the environment can be thought of as a combined dynamical system, in which the human motion is observable with multiple sensors. The skeleton data sensed by the Kinect and the hand motion data sensed by mobile inertial sensors are shared via Wireless LAN. This is a closed-loop system with two subsystems, as illustrated in Fig. 1. The human is subsystem 1 while the computing device system, including the mobile phone, the multiple sensors and the PC used for sensor fusion, can be treated as subsystem 2.

In subsystem 2, the phone can be seen as a moving target when the hand is moving. The user controls the moving of the phone. We can treat the phone as a flying machine, the input of which is the force of the hand. The motion of the phone is observed by multiple sensors. The trajectory is sensed by the Kinect sensor. Meanwhile, the orientation and the acceleration of the phone are observed by the built-in inertial sensors. The subsystem 2 is observable as we can determine the state of the system through the position observations and the acceleration measurements. This subsystem 2 is a time-delay system as the position is sensed by the Kinect, which has latency. The acceleration is sensed by the inertial sensors at a much higher sampling rate. We treat the acceleration as a non-delayed measurement. Our goal is to model this dynamical system with the GP prior method. The phone (hand) trajectory is defined by the movement of the user's muscles which drive nonlinear trajectories of the rigid body. The system we are modelling is a nonlinear dynamical system $g(x)$ with known inputs x and observed outputs y. At each time instant i, we get a measurement y_i, which is a function of the latent state x_i.

$$y_i = g(x_i) + \varepsilon_i, \tag{11}$$

where ε_i denotes Gaussian system noise.

In order to estimate the system state by fusing all the available observations including the positions and the accelerations, we need to illustrate the data availability in the sensor fusion system.

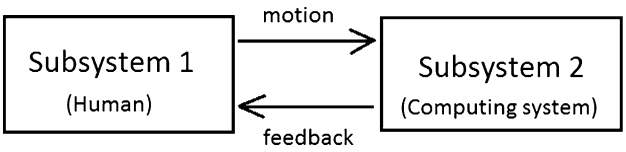

Fig. 1 Illustration of a closed-loop system with two subsystems including subsystem 1 (the human) and subsystem 2 (the computing system consists of the mobile phone, the multiple sensors and the PC)

3.2.2 *Data availability in the sensor fusion system*

Now we illustrate the data availability with Fig. 2. In order to illustrate the availability of sensor measurements at different time instants, we need to take account of the time delay (0.1 s) of the Kinect system.

In Fig. 2, we show the timing information and the delayed observations at $t = \frac{15}{90}$ s. The first row represents the timing information and the second row represents the acceleration measurements from the inertial sensors. In the third row, considering the effect of latency, the corresponding Kinect outputs are denoted as $p_i^d, i = 1, 2, 3, 4$. In the fourth row, it is shown that the actual available observations at $t = \frac{13}{90}$ s include 13 acceleration measurements and 2 position measurements, which are the noisy version of the system state (position) at $t = \frac{1}{90}$ s and $t = \frac{4}{90}$ s, respectively. We denote them as p_3 and p_4. The corresponding Kinect outputs become p_3^d and p_4^d, which are acquired at $t = \frac{10}{90}$ s and $t = \frac{13}{90}$ s, respectively.

3.2.3 *Autoregressive GP model*

Our proposed model is an autoregressive model, which acts like a moving "window". Gaussian Process regression is a

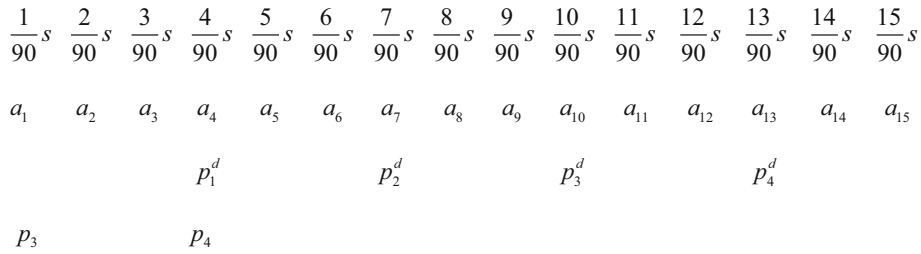

Fig. 2 Illustration of data availability: (1) The *first row* represents the time instants (90 Hz) (2) The *second row* represents the non-delayed acceleration measurements. (3) The *third row* represents the Kinect position measurements. They are the delayed noisy version of the system state (position). (4) Due to the 0.1 s latency, we assume that at $t = \frac{13}{90}$s, the available position measurements include p_1^d, p_2^d, p_3^d and p_4^d. p_3^d and p_4^d represent the delayed noisy version of the system state (position) at $t = \frac{1}{90}$s and $t = \frac{4}{90}$s

linear smoother [38] and the autoregressive Gaussian Process (ARGP) was applied for time series modelling in [12, 50]. In an ARGP of order L, the past L values $\mathrm{Y}_{(L)}$ are taken as the GP input while the output is y_t.

$$y_t = f\left(\mathrm{Y}_{(L)}\right) + \varepsilon_t, \tag{12}$$

where the GP function $f \sim GP(0, k)$ (k is the covariance matrix) and the white noise $\varepsilon_t \sim \mathcal{N}(0, \sigma^2)$, where σ is the standard deviation.

Here we put the ARGP model in a sensor fusion framework. The sensor observations are the inputs of the ARGP model. The problem is that the sensor observations include the delayed low-sampling-rate positions sensed by the Kinect and the high-sampling-rate accelerations measured by the inertial sensors. We want to build a GP prior model that incorporates these observations and takes into account the different noise characteristics of these sensors. We define the state of interest y_t as

$$y_t = f\left(p_{(L)}, a_{(l_a)}\right), \tag{13}$$

where y_t represents the GP predictive positions. The last L position measurements sensed by the Kinect are denoted as $p_{(L)}$, whereas $a_{(l_a)}$ are the last l_a acceleration measurements sensed by the inertial sensors, and $l_a = 3L + N_0 - 2$. The past L Kinect positions are the low-sampling-rate measurements in the assumed high-sampling-rate position space.

Considering the different sampling rates of these sensors, we have more acceleration measurements than position measurements. We define N_0 for alignment of delayed position and non-delayed acceleration. N_0 is a number that represents the latency between the Kinect position measurements and the acceleration measurements.

$$N_0 = \frac{dT}{\Delta t} = dT \cdot f_0, \tag{14}$$

where dT denotes the time delay (0.1 s) [1]. f_0 denotes the sampling rate of the inertial sensors, i.e. 90 Hz. Thus, $N_0 = 9$.

The graphical model for the GP sensor fusion is shown in Fig. 3. As defined in (13), every time the "window" takes the most recent L position measurements and the most recent l_a acceleration measurements. During the time period when the position measurements are unavailable, i.e. the most recent 0.1 s latency, the GPs make position prediction based on the most recent L position measurements and the most recent l_a acceleration measurements.

Now we have the state equation of the dynamical system, as defined in (13). Following this, we propose a novel Gaussian Process prior model for the dynamical system modelling. In our work, the human motion is relatively continuous and smooth in the trajectory-based target

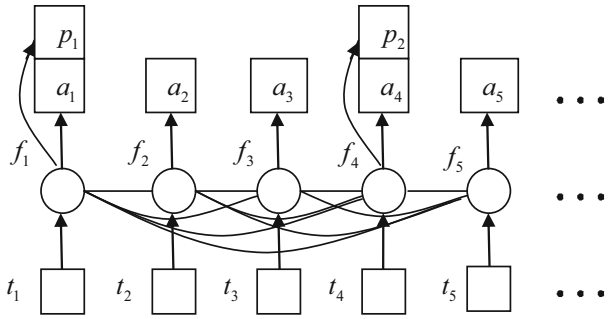

Fig. 3 Graphical model for the GP sensor fusion. The position p_i ($i = 1, 2, \ldots$) and acceleration a_j ($j = 1, 2, \ldots$). The sensors have different sampling rates (30 Hz and 90 Hz respectively, $dt = 1/90$ s). The higher frequency acceleration can augment the low-sampling-rate position. The Autoregressive GP model acts like a "slide window", taking the most recent L ($L = 5$) position observations and the corresponding l_a acceleration measurements, and giving the predictive positions

acquisition task. Here the covariance function chosen is a general smoother, the parameters of which are tuned to typical human motion. The parameters for the model are learnt from the training data using the maximum likelihood method. The GP model training was done offline, thus did not affect the performance of the online prediction. As GP regression is a linear smoother, the prediction is a linear combination of the training targets.

For sensor fusion with the GP prior model, the targets include the L positions and the l_a accelerations. If we can place an appropriate prior on the function space of the combination of position and acceleration, we can make position predictions based on the non-delayed accelerations during the 0.1 s. In order to find the joint distribution of the low-sampling-rate position P_{low} and the high-sampling-rate acceleration Acc_{high}, we apply the GP prior method and calculate an overall covariance matrix C_{all}, so

$$\begin{bmatrix} P_{low} \\ Acc_{high} \end{bmatrix} \sim \mathcal{N}(0, C_{all}). \tag{15}$$

So the following work is to apply GPs in a sensor fusion manner and find this joint distribution of the low-sampling-rate position and the high-sampling-rate acceleration with the GP prior method. Firstly, we discuss the GP prior prediction. Following this, we present the transformed GP priors and propose the novel and improved GP prior model for multi-rate sensor fusion, and give a detailed description on how to apply this model for fusing the Kinect sensor and inertial sensors.

3.2.4 GP prior model-based sensor fusion

The Gaussian Process prior framework can incorporate measurements and measurements of derivative

information, and allows GPs to perform sensor fusion of multiple observations in the form of multiple levels of derivatives of a measurand. In this paper, we further develop the work on GP priors in [33] by proposing a novel and improved GP prior model, which takes account of the different sampling rates and different noise characteristics of the sensors, and the Kinect latency in our problem.

Consider N observations of inputs X, i.e. the time instants (the time step is $\frac{1}{90}$ s) and outputs Y_{high}, i.e. the targets in the assumed high-sampling-rate position space, assuming Y_{high} are drawn from an N-dimensional normal distribution.

$$Y_{high} \sim \mathcal{N}(0, \Sigma), \tag{16}$$

where Σ is the $N \times N$ covariance matrix, the elements of which are functions of inputs X.

We denote the Kinect measurements as Y_{low}, which are the low-sampling-rate observations in the high-sampling-rate position space. $Y_{low} = [y_1, \ldots, y_n]^T$ is denoted as M_p, and the high-sampling-rate acceleration measurement $M_a = [a_1, \ldots, a_N]^T$.

Following this, we assume the observations $M = KY_{high}$, K is the transformation matrix. For the Kinect, the low-sampling-rate position measurements $M_p = K_p Y_{high}$, where K_p is defined in (17). For the mobile device, K_a is defined in (18), the acceleration measurements $M_a = K_a Y_{high}$ and $\Delta t = \frac{1}{90} s$.

$$K_p = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \cdots \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & \cdots \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & \cdots \\ & & & & \vdots & & & & & \ddots \end{bmatrix} \tag{17}$$

$$K_a = \frac{1}{\Delta t^2} \begin{bmatrix} 1 & -2 & 1 & & & & \\ & 1 & -2 & 1 & & & \\ & & \ddots & \ddots & & & \\ & & & 1 & -2 & 1 \end{bmatrix} \tag{18}$$

K_a is the classic second difference (derivative) operator (off-diagonal elements equal zeros). The connection between the low-sampling-rate positions and the high-sampling-rate accelerations can be expressed in an overall K_{all} matrix, which is defined in (19). By constructing an overall K_{all} matrix, we can build a Gaussian Process prior model for sensor fusion.

$$K_{all} = \begin{bmatrix} K_p \\ K_a \end{bmatrix}. \tag{19}$$

According to (15), we need to find the joint distribution of low-sampling-rate position and the high-sampling-rate acceleration. The GP training target M_{all} includes the position and acceleration.

$$M_{all} = \begin{bmatrix} M_p \\ M_a \end{bmatrix} = [p_{n-L+1}, \ldots, p_n, a_{l-l_a+1}, \ldots, a_l]^T, \tag{20}$$

where the most recent position p_n and the most recent acceleration a_l are acquired at the same time instant.

With the transformed GP prior method, we have this joint distribution

$$M_{all} = \begin{bmatrix} M_p \\ M_a \end{bmatrix} \sim \mathcal{N}\left(0, K_{all}\Sigma K_{all}^T + \begin{bmatrix} \Sigma_p & \\ & \Sigma_a \end{bmatrix}\right), \tag{21}$$

where the Σ_p and Σ_a represent the diagonal matrices of position and acceleration observation variances respectively (off-diagonal elements equal zeros). Σ_p has equal constants on the diagonal. Σ_a also has equal constants on the diagonal. We estimated these parameters by measuring the sensor noise characteristics. We determined the variance of the measurement noise through the sensor measurement of uncertainty illustrated with a histogram, and its Gaussian fit.

According to (22) and (23), we can calculate the conditional mean and variance of the predictive position P_{fusion} with GP sensor fusion method.

$$\mu_{2|1} = I_{l_a}\Sigma_{12}K_{all}^T\left(K_{all}\Sigma K_{all}^T\right)^{-1}M_{all}, \tag{22}$$

$$\Sigma_{2|1} = \Sigma_2 - I_{l_a}\Sigma_{12}K_{all}^T\left(K_{all}\Sigma K_{all}^T\right)^{-1}K_{all}\Sigma_{21}I_{l_a}^T, \tag{23}$$

$$P_{fusion} = I_{l_a}\Sigma_{12}K_{all}{}^T\left(K_{all}\Sigma K_{all}{}^T + \begin{bmatrix} \Sigma_p & \\ & \Sigma_a \end{bmatrix}\right)^{-1}M_{all}, \tag{24}$$

where I_{l_a} is the identity matrix of size l_a. P_{fusion} represent the predictive positions with the sensor fusion approach. Σ_{12} represents the covariance matrix between the training inputs and the test inputs, whereas Σ denotes the covariance matrix between the training inputs and themselves. The Σ_p and Σ_a represent the diagonal matrices of position and acceleration observation variances (off-diagonal elements equal zeros) respectively. Σ is a $l_a \times l_a$ matrix. K_{all} is a $(L + l_a) \times l_a$ matrix as K_p is a $L \times l_a$ matrix in the form of (17) and K_a is a $l_a \times l_a$ matrix in the form of (18).

Algorithm 1: Illustration of how the proposed Autoregressive GP model works

1. Measure the acceleration with mobile inertial sensors and the Kinect position.
2. The number of the assumed non-delayed position (the fourth row in Figure 2) is denoted by n. According to n, construct K_p and K_a accordingly (see (17) and (18)).
   When $n < L$, adjust K_p and K_a accordingly (This n equals a smaller L. Replace L with n in (20)). K_p is a $n \times (3 \cdot n + N_0 - 2)$ matrix and K_a is a $(3 \cdot n + N_0 - 2) \times (3 \cdot n + N_0 - 2)$ matrix, where $(n = 1, \cdots, L-1)$.
   When $n \geq L$, K_p and K_a are both fixed matrices.
3. Construct the target vector M_{all} according to (20).
4. Start the GP fusion. Make prediction according to (22) and (23).

3.3 Use of GP model to fuse Kinect and inertial sensors

We illustrate how to take the proposed sensor fusion approach to predicting the position in the Algorithm 1. In Experiment 1 in Sect. 5, we used $L = 5$. Now we illustrate how to construct a K_{all} matrix when $n = 2 < L$. From (17), (18) and (19), we know how to construct the overall K_{all} matrix. How does it relate to the measurements including the positions and the accelerations here? In Fig. 2, by the time we have two observations p_3^d and p_4^d, we have 13 accelerations in the meantime. Here K_p is a 2×13 matrix and K_a is a 13×13 matrix. Thus,

$$K_{all,L=2} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \frac{1}{\Delta t^2} & \frac{-2}{\Delta t^2} & \frac{1}{\Delta t^2} & & & & & & & & & & \\ & \ddots & & & & & & & & & & & \\ & & & & & \ddots & & & & & & & \\ & & & & & & & & & & \ddots & & \\ & & & & & & & & & \frac{1}{\Delta t^2} & \frac{-2}{\Delta t^2} & \frac{1}{\Delta t^2} \end{bmatrix}. \tag{25}$$

From (25), we can see that there are $3L + 7$ accelerations (13 here) when there are L position observations (2 here). There are 9 more accelerations $(a_5, \ldots, a_{13}$ here).

$$K_{all,L=2} Y_{t=\frac{13}{90}s} = M_{all,L=2}, \tag{26}$$

where $Y_{t=\frac{13}{90}s}$ is the assumed high-sampling-rate position observations and $M_{all,L=2} = \left[p_3^d p_4^d a_1 a_2 \ldots a_{13}\right]^T$ are the targets including the low-sampling-rate positions and the high-sampling-rate accelerations. If we set the test inputs to $t = \frac{5}{90}s, \ldots, \frac{13}{90}s$, the prediction is based on $a_5, \ldots, a_{13}$ during the test inputs period and we get 9 predictive positions with GPs according to (24).

4 System overview

The equipment includes Microsoft Kinect, SHAKE SK7[1] [57] and Nokia N9. The frame rate of the Kinect sensor is 30 Hz, whereas the sampling rate of inertial sensors is much higher, 90 Hz. The OpenNI drivers and the motion tracking middleware (NITE) are used [35].

4.1 Augmenting the Kinect system with a mobile device in a spatially aware display

The system architecture of our spatially-aware display application is shown in Fig. 4. Our design focuses on a 2D (the vertical XY plane) version of a spatially-aware display. We aim to test whether a sensor fusion system improves user performance by designing a trajectory-based target acquisition task with this system.

An example application is shown in Fig. 5. A user is exploring the digital information stored in the physical space. By moving his phone to different locations following a trajectory, he can perform a target selection task. Imagine this is a virtual bookshelf application [34]. We store different digital books in different targets' locations. Then the user can search and browse those digital books by category. Since accurate positioning and feedback is critical to this application, we need to deal with the Kinect position jitter. The lags and overshoot problems occur when we apply a position-only Kalman filter for filtering the fast jitter on the hand movement data. Our novel GP sensor fusion method helps reduce the lags and deal with the overshoot problems when the user tries to find the target by moving the hand quickly.

The Nokia N9 is equipped with a 3-axis accelerometer, which can be used to estimate the hand acceleration. The Kinect is put on a table. The user stands in front of the Kinect with the N9 held in the right hand, and is directly

[1] http://code.google.com/p/shake-drivers/.

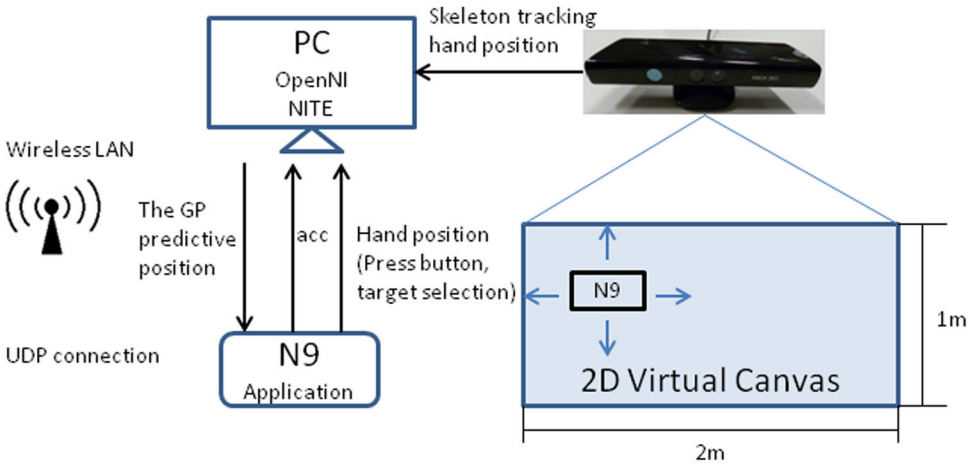

Fig. 4 System architecture. A Wireless LAN is used for UDP connection. The OpenNI and NITE middleware are used. The Kinect senses the hand position and sends it to the PC. The accelerometer data from the phone is also sent to the PC. Our novel GP sensor fusion model is applied for fusing the position and the acceleration. The GP predictive position is sent to the phone. The phone is a movable window on the 2D virtual canvas, on which we put a pre-designed trajectory and 6 targets. When the virtual button on the phone screen is pressed, the target on the canvas is selected and the current hand position is sent back to the PC

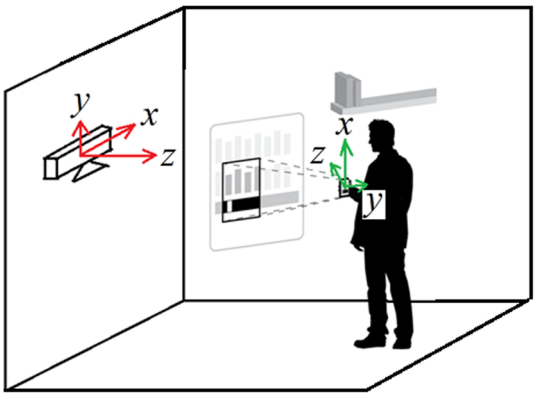

Fig. 5 Spatially aware display application. A phone user performs a trajectory-based target selection task in 2D space

facing the *XY* plane, i.e. the vertical interaction plane.[2] The information (the trajectory and the targets) is spread out on a flat virtual space. The phone acts as a movable window (size 48 mm $\times$ 86 mm) on this much larger 2D virtual canvas. The 2D canvas covers a 2 m $\times$ 1 m area.

When the user moves the hand in the 2D plane in front of the Kinect, he/she tries to explore the digital information stored in the physical space. The trajectory and the targets are located on the virtual canvas, which is shown in Fig. 11. There is a mapping between the 2D real world space (mm) and the 2D virtual canvas (pixels). In our application, 1 mm $\times$ 1 mm = 10 pixels $\times$ 10 pixels. Along the x-axis, the range is $(-1000, 1000)$ mm, whereas (0, 1000) mm for the y-axis.

[2] In this implementation a fixed rotation matrix between the phone body frame and the Kinect frame is assumed.

The 2D plane is like a big virtual canvas, and the phone screen is a small movable window, through which we can see a part of the virtual canvas. The hand position (x, y) indicates the phone position. When the hand moves, the Kinect and the accelerometers sense this, and the predictive position from the GP sensor fusion is sent to the phone to update the display on the phone screen. The user needs to move the hand along the trajectory. When a target appears on the screen, the user performs target selection. A selection occurs when the virtual button on the phone screen is tapped. On the N9, we designed a square virtual button and put it at the right side of the screen as shown in Fig. 10. Whenever the user presses the button, the phone will send a signal and the PC will record the current hand position.

In the augmented system, a Wireless LAN is used for data transmission. The hand tracking positions sensed with the Kinect and the accelerometer data from the N9 are sent to the PC via WiFi. The position measurements and the accelerations are fused with our proposed GP model method for position prediction. The data transmission between the phone and the PC includes three parts:

1. The phone transmits the accelerometer data to the PC.
2. Sensor fusion with our novel GP model on the PC. The PC sends the GPs predictive position (x, y) to the phone.
3. The phone sends a signal to the PC when the user presses the virtual button to select the target.

This can be seen in Fig. 4.

Four coordinate systems are involved in our sensor fusion system. (1) Earth's North-East-Down (NED) frame (e): this is SK7's reference frame. (2) Kinect frame (k): the joint's 3D coordinates are expressed in this coordinate

system. (3) SK7 / N9 body frame (b). (4) N9 phone image frame (i): The top left corner is (0, 0) (pixels) in the landscape mode. A detailed description on how to estimate the acceleration through inertial sensor fusion can be found in [7]. In this paper, we focus on how to use the proposed GP prior model to fuse the Kinect position and the acceleration measured by mobile inertial sensors.

5 Experiments

5.1 Experiment 1 : sensor fusion

We conducted an experiment to test the performance of the proposed GP prior model-based sensor fusion system. In this experiment, we used a leap motion controller to sense the hand position (90 Hz). The V2 Tracking Beta SDK provides the hand tracking with high accuracy and near-zero latency [31]. This was used as the baseline for evaluating the performance of the GP sensor fusion method. Meanwhile, we collected the hand position data sensed by the Kinect and the hand acceleration measured by the mobile inertial sensors. We compared the sensor fusion approach with the position-only Kalman filter prediction method and the position-only GP, and concluded that the GP prior model-based sensor fusion is superior to the two methods. The proposed approach helps improve the accuracy of position estimation and reduce the lag.

5.1.1 Experiment design

Before starting the experiment, we calibrated the position tracking systems including the Leap Motion Controller and the Kinect sensor. The inertial sensors were also calibrated. We aligned the Kinect frame and the Leap Motion tracking frame, and analysed the hand movement along the x-axis as an example. In this way, the two frames have the same origin along the x-axis in the space.

In this experiment, the user's right hand motion was sensed by the Leap Motion Controller, the Kinect and the inertial sensors pack. The user put the hand above the Leap Motion Controller (the height is approximately 20 cm), and performed a hand movement with a mobile device (SK7) held in the hand in the Kinect field of view. The distance between the Kinect and the Controller is 1.5 m. At the beginning, the user put the hand above the controller, then moved the hand along the $+x$-axis (the distance is approximately 20 cm) and then stopped. The process took 2 s.

5.1.2 Experimental method

In this experiment, we test the GP prior model-based sensor fusion approach. We chose $L = 5$ as this can give a good prediction result and is very computationally efficient. For a constant sampling rate (90Hz), the covariance matrix is a fixed matrix (27×27) (20). We built a position-only Kalman filter, which uses a continuous Wiener process acceleration model as discussed in [7]. This position-only KF makes 1 step ($\frac{1}{30}$ s) prediction first, then the Kinect position measurement is used to update the system state. Based on the updated state, this KF makes 3 steps ahead prediction to deal with the 0.1 s delay. We also compared the GP sensor fusion with the position-only autoregressive GP method, which uses the most recent L position measurements for multi-step ahead prediction. As there is a 0.1 s delay and the sampling rate of the Kinect is 30 Hz, the position-only GP makes 3 steps prediction. The position-only GP and the GP sensor fusion use the same hyperparameters, the maximum likelihood estimate of which can be calculated using the time-stamped human motion training data and the standard optimisation algorithm. We collected and used the time-stamped position measurements (10 s, 300 data-points) sensed by the Kinect as the training dataset.

The uncertainty of Kinect position measurements is measured to be (SD) $\sigma = 8$ mm. The uncertainty of the acceleration estimation in the Kinect system is measured to be (SD) $\sigma_a = 100$ mm/s^2. The GP hyperparameters are set to $v_0 = 5.66 \times 10^4$, $\omega_1 = 4.19$, $\sigma_y^2 = 64$ and $\sigma_a^2 = 100^2$.

5.1.3 Experimental results

Measurements

In the experiment, the Kinect sensed the hand position. The hand acceleration was measured by mobile inertial sensors held in the hand. The hand position sensed by the leap motion controller was used as the baseline. Figure 6 illustrates the x-axial position measurements (in the upper panel) and the corresponding x-axial acceleration measurements (in the lower panel). We can see that the Kinect position measurements are noisy and delayed. The GP sensor fusion is to fuse the noisy, delayed low-sampling-rate position observations and the higher frequency acceleration measurements with the proposed GP prior model.

Sensor Fusion and Comparison In this part, We fuse the Kinect position observations and the acceleration measurements with the GP prior model-based sensor fusion approach. We compare it with the position-only KF and the position-only GP.

1. The position-only Kalman filter prediction

Figure 7 shows 3 signals, including (1) the baseline data, (2) the position measurements and (3) the predictive positions with the position-only KF. We analysed the accuracy of the position predicted with this position-only KF by comparing the prediction results with the baseline data. The results are summarised in Table 1.

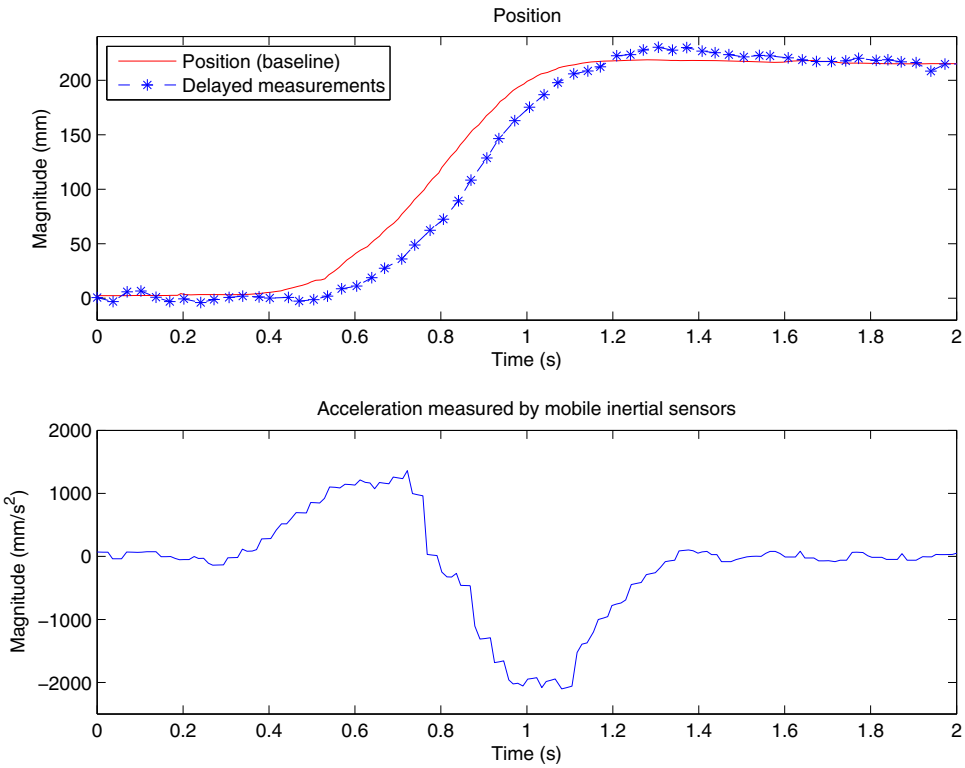

Fig. 6 *Upper panel*: The x-axial position measurements and the baseline data. *Lower panel*: x-axial acceleration estimated with inertial sensors and expressed in Kinect coordinate system

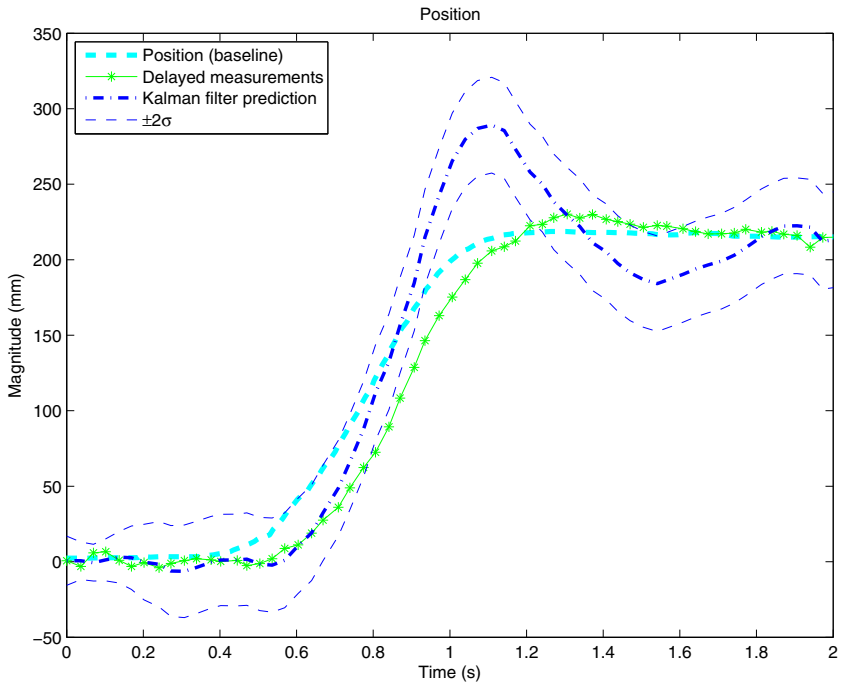

Fig. 7 The position-only Kalman filter prediction. Plots show the mean $\pm 2\sigma$. The figure shows 3 signals: (1) the baseline data (2) the position measurements (2) the predictive positions with the position-only KF. Plots show the mean $\pm 2\sigma$

2. Comparison with the position-only GP

In addition to the position-only KF, we also compare the GP sensor fusion with the position-only GP. The experimental results are shown in Fig. 8, which shows 4 signals, including (1) the baseline data, (2) the Kinect position measurements, (3) the position-only GP prediction result and (4) the predictive positions with the GP sensor fusion method. We use the method described in the Algorithm 1. We can see that the position prediction with the GP sensor fusion is smoother in comparison with the position-only GP

result. Besides, the uncertainty of position prediction with the GP sensor fusion is much smaller. Moreover, the system lag is reduced with the GP sensor fusion approach. This proves that the high-sampling-rate acceleration can compensate for the effect of position uncertainty and lag in the Kinect system.

3. Accuracy of position estimation

In order to analyse the accuracy of the mean position prediction, we calculate the RMSE based on the baseline data. For the KF, the position-only GP and the GP sensor fusion approach, this RMSE is the root of the average of the squares of the difference between the mean predictive positions and the baseline data. We compare the GP sensor fusion approach with the position-only KF prediction and the position-only GP prediction method. The results are summarised in Table 1.

In comparison with the baseline position data, the RMSE of the noisy and delayed position measurements sensed by the Kinect is 19.75 mm. The measured uncertainty is 8 mm. The RMSE of the mean position predicted by the position-only KF is 29.19 mm. The uncertainty (standard deviation SD) after convergence is 15.84 mm. The RMSE and uncertainty of the mean position predicted with the GP approaches are illustrated in Table 1. We can see that the sensor fusion with GP helps reduce the error of mean position prediction and the uncertainty of the prediction. In comparison with the position-only GP, the RMSE of the mean position prediction is reduced by 35.8 % and the uncertainty of the mean position prediction was reduced by 59.7 %.

Table 1 Comparison of accuracy—compare the GP sensor fusion approach with the position-only KF and the position-only GP method

Methods	Accuracy (mm)	
	RMSE of mean prediction	SD (σ)
Position-only KF	29.19	15.84
Position-only GP	10.76	29.89
GP sensor fusion	6.91	12.04

Thus, the proposed approach is superior to the position-only KF and the position-only GP. As the KF is a special case of a GP and the proposed approach can be put in a KF framework and implemented by carefully designing a customised variant of the multi-rate KF, there is no need to compare the proposed approach with a sensor fusion-based KF. We conclude that the proposed approach helps improve the accuracy of the position estimation.

4. Lag reduction

Now we use the unbiased estimate of the cross-correlation function to analyse the time delay between the GP predictive position signal and the Kinect position measurement signal. Figure 9 shows the cross-correlation sequence in a length 359 vector, where the GP predictive position signal and the Kinect measurement signal are both vectors of length 180 (interpolation 90 Hz), respectively. The peak was acquired at 190. Thus, the lag was reduced by 0.11 s.

5.1.4 Summary on experiment 1

In this experiment, we tested the proposed GP prior model-based sensor fusion approach. The sensor fusion with the

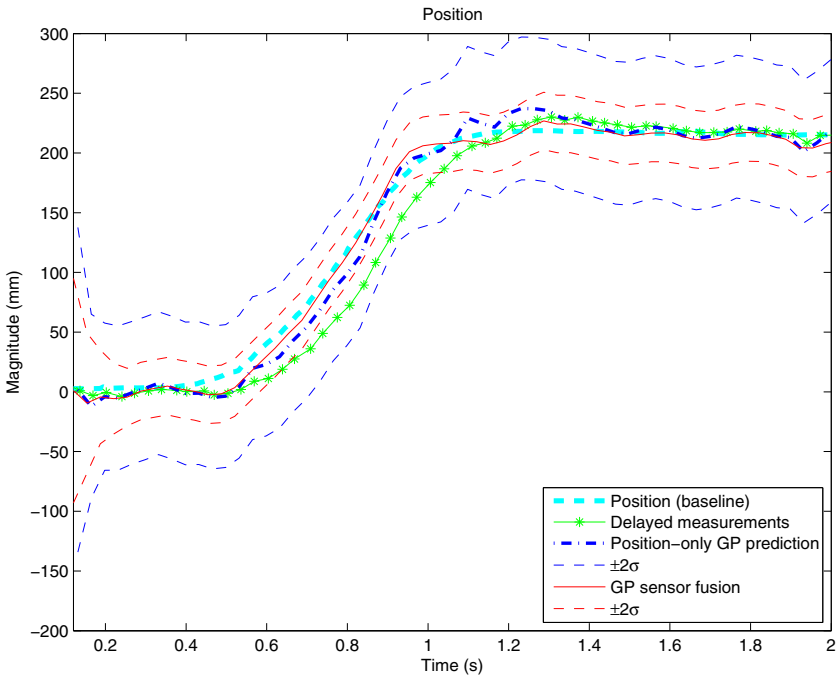

Fig. 8 Comparison of position-only GP and sensor fusion with GP (L = 5). Plots show the mean $\pm 2\sigma$. The figure shows 4 signals: (1) the baseline data (2) the position measurements (3) the position-only GP prediction (4) the prediction with the GP sensor fusion

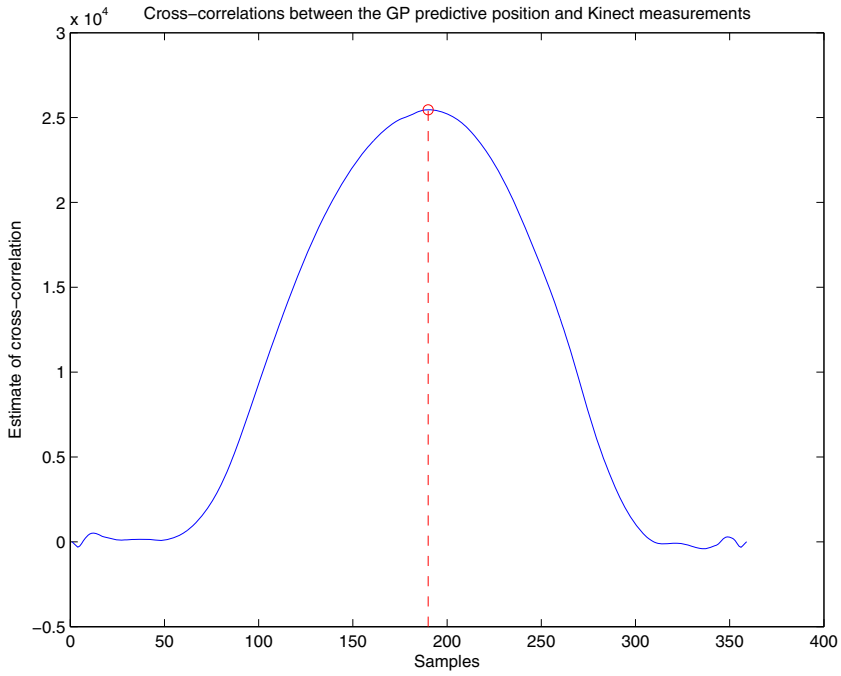

Fig. 9 The GP sensor fusion helps reduce the lag. Plots show the cross-correlation sequence. The peak was acquired at 190. Thus, the lag was reduced by 0.11 s

proposed GP prior model helps improve the accuracy of position estimation, and reduce the lag of the conventional Kinect system by 0.11 s.

5.2 Experiment 2: user study—trajectory-based target acquisition task

Our user study aims to test our sensor fusion system when the user performs a 2D trajectory-based target selection task in a spatially aware display application.

5.2.1 Participants and apparatus

There were 12 participants in total (6 male, 6 female). They were aged between 20 and 35 years (mean age 28). Participants were recruited by email, and some volunteered from the academic community in our school. The task was performed on a Nokia N9, which is a phone with 3.9 inches display (480 pixels $\times$ 854 pixels or 48 mm $\times$ 86 mm).

5.2.2 Data collection and analysis

We aim at analysing the accuracy of target selection and the task completion time. In the task, we recorded the hand position sensed by the Kinect and the hand acceleration measured by the Nokia N9. When the participant performed the target selection task, the hand position was recorded. We analysed the accuracy of target selection. Besides, we measured and analysed the task completion time. Following the experiment, the participants completed the NASA Task Load Index [16] questionnaire, which gathered subjective assessment of usability of the system.

5.2.3 Experiment design

The participants were instructed to interact with the system in a comfortable way. Then they were instructed to perform a trajectory-based target selection task as accurately and quickly as possible. Each participant performed the task in (1) the Kinect system (2) the sensor fusion system. After each session, the user completed the questionnaire. The users were not informed which system they were using. Task 1 and task 2 were denoted on the questionnaire.

At the beginning of the experiment, the user stood in front of the Kinect with a mobile device (Nokia N9) held in the hand and was directly facing the *XY* plane, i.e. the vertical interaction plane. Once skeleton tracking locked on, the user moved his hand following the pre-designed trajectory, which was only shown on the phone screen. No visual information is present in the real world outside the device's display. Whenever a target appeared on the trajectory, the user selected it by pressing the virtual button on the phone screen. Meanwhile, this position was recorded and sent back to the PC. It was compared with the ground truth data (we know the real position for the targets) for error rate analysis. This can be seen in Fig. 10.

5.2.3.1 Trajectory design We used a combination of a straight line and a square wave curve for modelling the trajectory for the target selection task. Six targets were

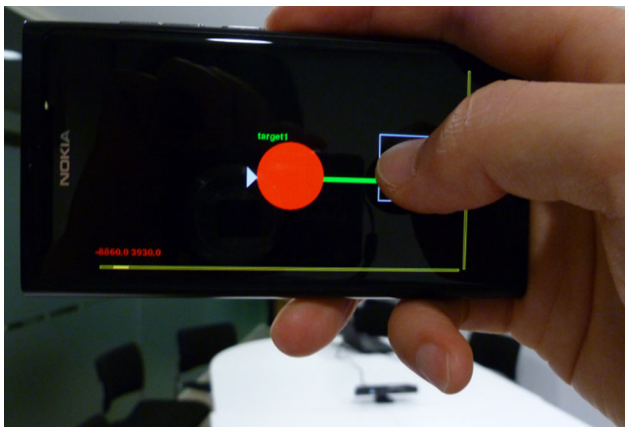

Fig. 10 The interface on the Nokia N9 phone screen in the spatially aware display application. A user was performing the trajectory-based target selection task. The first target was shown on the screen. The *square box* on the *right* of the screen is the virtual button. When the button is pressed, the target is selected. Meanwhile, visual feedback (the *color* of the button changes) is provided for the user during the target selection task

located along the trajectory. The pre-design trajectory and the targets are shown in Fig. 11.

5.2.4 Experimental method

The Kinect senses the position of the hand and the N9 accelerometer measures the hand acceleration. The accelerometer embedded in the N9 was calibrated before the experiment started. When the PC receives the Kinect position and the acceleration sensed by the N9, the GP prior model is applied for sensor fusion. We use the same setting ($L = 5$) as in Experiment 1. The GP predictive positions, i.e. (x, y) mm coordinates, are sent to the phone for updating the canvas display. The predictive hand position is treated as the position of the screen centre. Thus, the digital content (e.g. a part of the trajectory) located in this area can be displayed on the screen. We compared our system with the conventional Kinect system, in which a position-only Kalman filter [7] that uses a continuous Wiener process acceleration model was applied for filtering the noisy position measurements. The filtered position was sent to the phone for updating the canvas display. We compared this Kinect system with our sensor fusion system.

5.2.5 Experimental results

Accuracy of target selection The target selection accuracy is a subjective measurement. When the user presses the button, the recorded position is the place where the user believes the target is located. We compared the target selection position with the ground truth data, i.e. the real target position defined on the virtual canvas. In order to compare the accuracy of target selection in two systems, we calculated mean square error (MSE) and the root mean square error (RMSE).

The comparison results are shown in Fig. 12. The MSE of target selection in the Kinect system is 3.7263×10^5 pixel^2 (SD 2.1096×10^5). For the sensor fusion system, it is 2.2975×10^5 pixel^2 (SD 1.2452×10^5). The MSE is reduced by 38.3 %. The RMSE of target selection in the Kinect system is 610.44 pixel. For the sensor fusion system, it is 479.32 pixel. The RMSE is reduced by 21.5%.

Results were analysed using a repeated measures Analysis of Variance (ANOVA). The sensor fusion system has a statistically significant effect on the target selection accuracy, $F(1, 11) = 10.86$, $p = 0.0071$.

Task completion time

The task completion time for our sensor fusion system (M = 32.41 s, SD = 12.04 s) is shorter than that for the

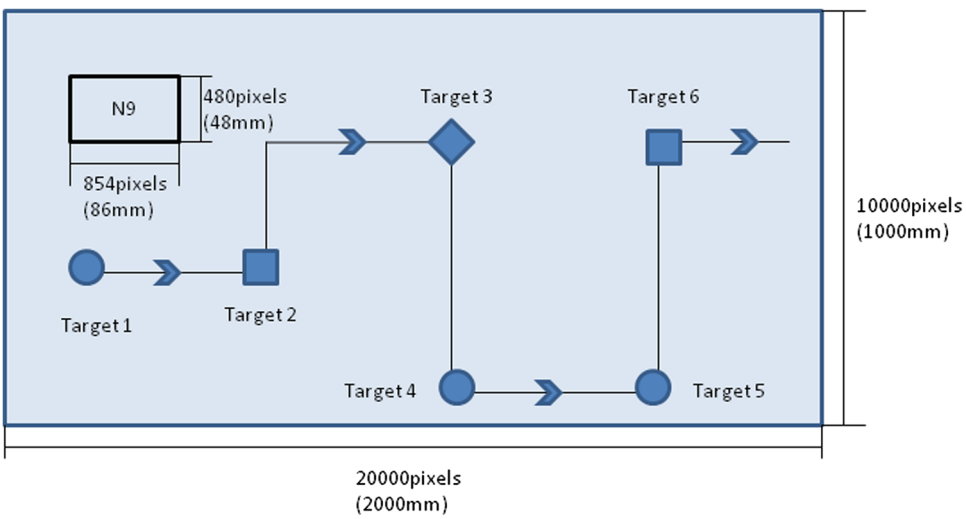

Fig. 11 2D virtual canvas design. The canvas covers a 2 m × 1 m area in the Kinect *XY* plane. N9 is a phone with 3.9 inches display (480 pixels × 854 pixels) (size 48 mm × 86 mm). Thus, when the size of the canvas is expressed in pixels, it is 20,000 pixels × 10,000 pixels. We use the *straight line* and *square wave* for modelling the trajectory, on which 6 targets are located

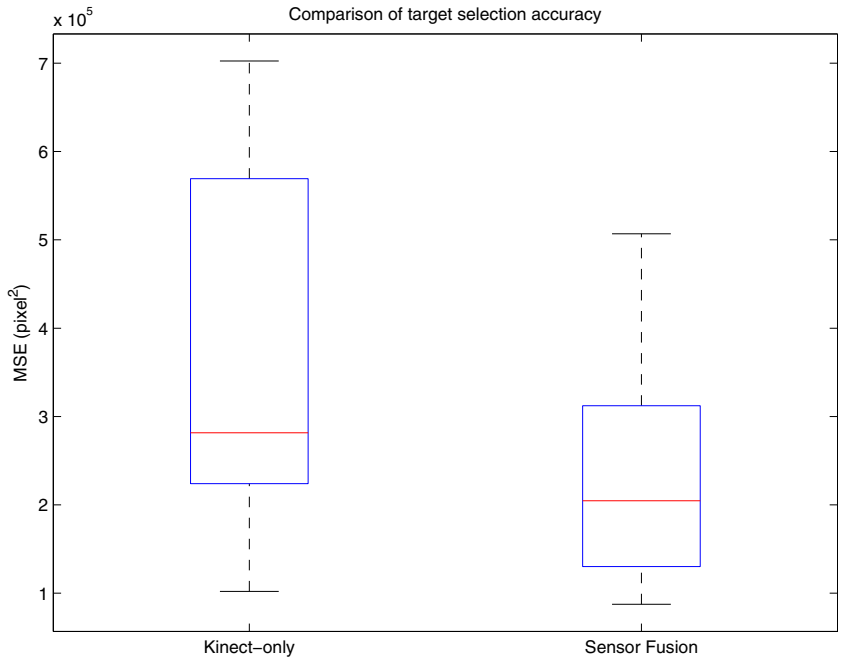

Fig. 12 Comparison of target selection accuracy. *Left column*: the Kinect system. *Right column*: GPs sensor fusion system. It can be seen that the target selection error is reduced by the sensor fusion

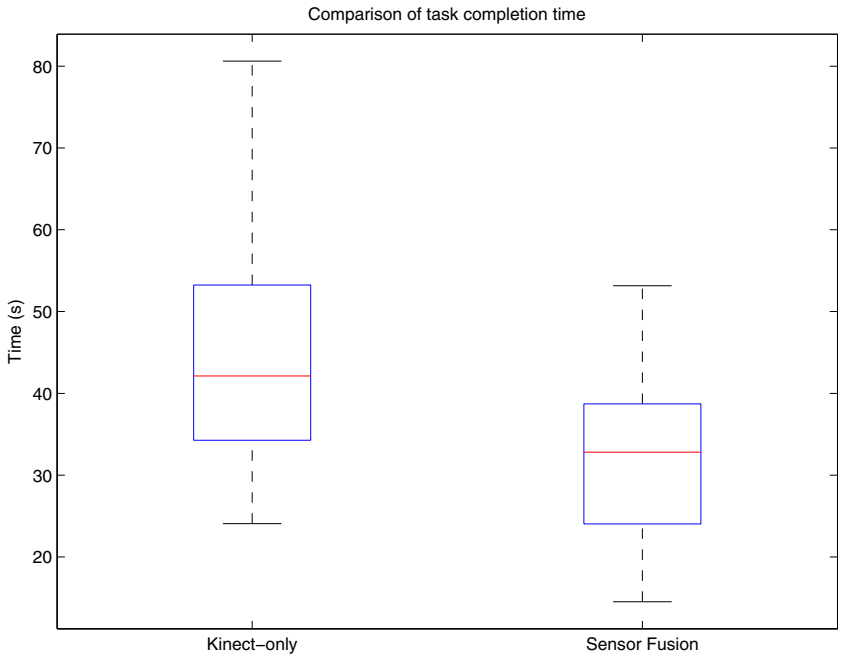

Fig. 13 Comparison of task completion time. *Left column*: the Kinect system. *Right column*: GPs sensor fusion system. It can be seen that the average task completion time is reduced by the sensor fusion

Kinect system (M = 44.21 s, SD = 14.77 s). The average task completion time is reduced by 26.7%. A comparison of the average task completion time is shown in Fig. 13.

Results were analysed using a repeated measures analysis of variance (ANOVA). The GPs sensor fusion system has a statistically significant effect on the task completion time, $F(1, 11) = 12.05$, $p = 0.0052$.

Questionnaire

Following each session of the experiment, each participant was asked to complete the NASA Task Load Index questionnaire. For each scale, the line is divided into 20 intervals. From left (low) to right (high), scores range from 0 to 20 [45]. A lower score indicates a better performance. The conventional Kinect system obtained a score of 619,

whereas our sensor fusion system obtained a score of 513. The subjective load varied in line with the objective measures of speed and accuracy.

For each scale, we calculated the mean score and the standard deviation. The results are shown in Table 2. We can see that the average subjective assessment of usability of our sensor fusion system is better than that of the Kinect system.

The comparison results of the NASA Task Load Index for the Kinect system and the sensor fusion system are shown in Fig. 14. The lower score of each scale indicates a better performance of the system. In Fig. 14, the Boxplot shows the distribution of each scale data for two systems.

We have two systems and need to do a paired sample test. Results were analysed using a Wilcoxon signed-rank test. We get the following results: (1) The mental demand, $p = 0.0137$. (2) The physical demand, $p = 0.0898$. (3) The temporal demand, $p = 0.0508$. (4) The performance, $p = 0.0249$. (5) The effort, $p = 0.1611$. (6) The frustration, $p = 0.0195$. It can be seen that the GPs sensor fusion system has a statistically significant effect on the mental demand, the temporal demand, the performance and the frustration. Thus, the sensor fusion system outperforms the Kinect system in the subjective assessment of usability of the system.

Table 2 The NASA Task Load Index

Scale	Scores for different systems			
	Kinect system		Sensor fusion system	
	Mean	SD	Mean	SD
Mental demand	7.17	3.64	5.08	2.78
Physical demand	7.75	3.11	7.17	3.54
Temporal demand	11.25	3.47	9.92	4.01
Performance	6.50	4.06	5.25	3.08
Effort	10.92	4.66	9.33	4.38
Frustration	8.17	4.37	6	4.11

5.2.6 *Summary on experiment 2*

Experimental results show that our system enables the user to perform the task more accurately and more quickly in comparison with the Kinect time-delay system. The target selection error and the task completion time are both reduced by the GP sensor fusion. Moreover, the participants reported improved performance in our system.

6 Discussion

The proposed GP prior model-based sensor fusion method was used to fuse the position sensed by the Kinect and the acceleration measured by the mobile device in this paper. We built this Kinect-augmented system to test the proposed GP sensor fusion approach, which can be generalized for a wider range of applications. The idea is to change the transformation matrix K in (7). In this paper, the transformations are limited to approximations of derivative transformations, so the K was set as the classic second derivative operator, i.e. K_a in (18). As long as we have this transformation matrix K, we can find the corresponding

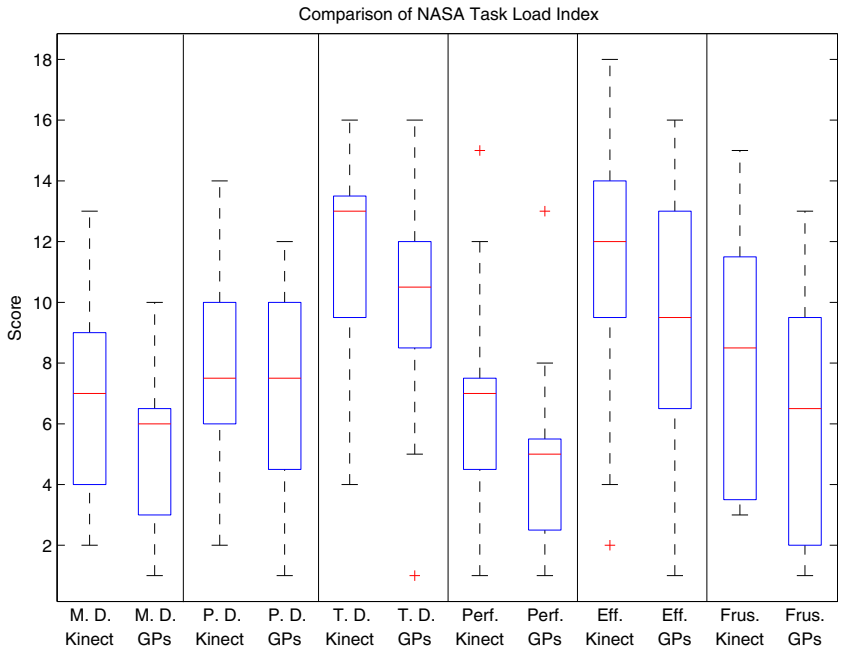

Fig. 14 The *Boxplot* shows the comparison results of the NASA Task Load Index for the Kinect system and the sensor fusion system. The 6 scales along the x-axis are (1) Mental demand (2) Physical demand (3) Temporal demand (4) Performance (5) Effort (6) Frustration. The "Kinect" (along the x-axis) represents the Kinect system. The "GPs" (along the x-axis) represents the sensor fusion system. A lower score indicates a better performance

distribution of $M = KY$ based on Y. This provides us a feasible way to fuse data from multiple sources, as discussed in Sect. 3.1.2.

In this way, we can fuse multiple observations that might be a mixture of readings from different physical sensors or different operators applied to the data, to derive a model based on a latent variable, which is compatible with all of them.

7 Conclusion

This paper presents a novel GP prior model-based sensor fusion approach to modelling sensor fusion system. The interaction system in our work improves the accuracy of the skeleton joint position estimation and reducing the lag by fusing the Kinect and the built-in inertial sensors in a mobile device. The proposed novel and improved GP prior model incorporates the low-sampling-rate position measurements and the higher frequency acceleration, taking the different noise characteristics of these sensors into account.

This type of sensor fusion system is of great benefit for location-aware applications. Firstly, the sensor fusion can improve the quality of inferred joint positions, as the high-sampling-rate acceleration signal can augment the low-sampling-rate, noisy position measurements. It can also help to reduce the lag, as the inertial sensing has a lower latency than the position sensed by the Kinect.

We conducted two experiments to test the GP prior model-based sensor fusion system. Experimental results show that the GP sensor fusion helps improve the accuracy of position estimation, and reduce the lag (0.11 s). In the second experiment, we built a spatially aware display application for user study. The user performed the trajectory-based target acquisition tasks in two different systems: (1) the Kinect system; (2) the sensor fusion system. In comparison with the Kinect system, the user performed the trajectory-based target acquisition task more quickly and more accurately in our sensor fusion system. The average task completion time was reduced by 26.7 % and the MSE of target selection was reduced by 38.3 %. We used the NASA Task Load Index to analyse the subjective assessment of usability of the system. The experimental results show that the GPs sensor fusion system has a statistically significant effect on the mental demand, the temporal demand, the performance and the frustration. We conclude that the GP prior model-based approach helps improve the user performance in the sensor fusion system. Moreover, we generalize the proposed approach and discuss that the GP prior model-based sensor fusion has the potential to be used in a wider range of sensor fusion systems.

Acknowledgments The authors would like to thank all of the experiment participants for their time and valuable feedback, and also thank Dr. Simon Rogers and Dr. John Williamson for their helpful discussions and valuable suggestions. This research has been jointly funded by University of Glasgow and China Scholarship Council. Nokia donated some of the equipment used.

References

1. Azimi M (2012) Skeletal joint smoothing white paper. http://msdn.microsoft.com/en-us/library/jj131429.aspx. Accessed Aug 2014
2. Azuma RT et al (1997) A survey of augmented reality. Presence 6:355–385
3. Bo A, Hayashibe M, Poignet P et al (2011) Joint angle estimation in rehabilitation with inertial sensors and its integration with Kinect. In: EMBC'11: 33rd annual international conference of the IEEE engineering in medicine and biology society, pp 3479–3483
4. Casiez G, Roussel N, Vogel D (2012) 1 € filter: a simple speed-based low-pass filter for noisy input in interactive systems. In: Proceedings of the 2012 ACM annual conference on human factors in computing systems, ACM, pp 2527–2530)
5. Conner B, Holden L (1997) Providing a low latency user experience in a high latency application. In: Proceedings of the 1997 symposium on Interactive 3D graphics, ACM, pp 45–ff
6. Corke P, Lobo J, Dias J (2007) An introduction to inertial and visual sensing. Int J Robot Res 26:519–535 (SAGE Publications)
7. Feng S, Murray-Smith R (2014) Fusing Kinect sensor and inertial sensors with multi-rate Kalman filter. In: IET conference on data fusion target tracking 2014: algorithms and applications (DF TT 2014), pp 1–8
8. Feng S, Murray-Smith R (2016) Transformations of Gaussian Process priors for user matching. Int J Hum Comput Stud 86:32–47 (Elsevier)
9. Fitzmaurice GW (1993) Situated information spaces and spatially aware palmtop computers. Commun ACM 36:39–49 (ACM)
10. Forrester A, Sobester A, Keane A (2008) Engineering design via surrogate modelling: a practical guide, Wiley
11. Forrester AI, Sóbester A, Keane AJ (2007) Multi-fidelity optimization via surrogate modelling. In: Proceedings of the royal society of london a: mathematical, physical and engineering sciences, vol 463. The Royal Society, pp 3251–3269
12. Girard A, Rasmussen CE, Candela JQ, Murray-Smith R (2003) Gaussian process priors with uncertain inputs—application to multiple-step ahead time series forecasting. In: Becker STS, Obermayer K (eds) Advances in neural information processing systems, vol 15. MIT Press, Cambridge, pp 529–536
13. Goovaerts P (1998) Ordinary cokriging revisited. Math Geol 30:21–42
14. Hall DL, Llinas J (1997) An introduction to multisensor data fusion. Proc IEEE 85:6–23 (IEEE)
15. Han Z-H, Görtz S, Zimmermann R (2013) Improving variable-fidelity surrogate modeling via gradient-enhanced kriging and a generalized hybrid bridge function. Aerosp Sci Technol 25: 177–189 (Elsevier)
16. Hart SG, Staveland LE (1988) Development of NASA-TLX (task load index): results of empirical and theoretical research. Hum Ment Workload 1:139–183 (Amsterdam, Holland)
17. Hennessey C, Noureddin B, Lawrence P (2008) Fixation precision in high-speed noncontact eye-gaze tracking. Syst Man Cybern Part B Cybern IEEE Trans 38:289–298 (IEEE)

18. Hol J, Schön T, Luinge H, Slycke P, Gustafsson F (2007) Robust real-time tracking by fusing measurements from inertial and vision sensors. J Real Time Image Process 2:149–160 (Springer)
19. Jeon S, Tomizuka M, Katou T (2009) Kinematic Kalman filter (KKF) for robot end-effector sensing. J Dyn Syst Meas Control 131:021010
20. Khaleghi B, Khamis A, Karray FO, Razavi SN (2011) Multisensor data fusion: a review of the state-of-the-art. Inf Fusion, Elsevier
21. Ko J, Fox D (2009) GP-BayesFilters: Bayesian filtering using Gaussian Process prediction and observation models. Auton Robots 27:75–90 (Springer)
22. Ko J, Klein DJ, Fox D, Haehnel D (2007) GP-UKF: unscented Kalman filters with Gaussian Process prediction and observation models. In: Intelligent robots and systems, 2007. IROS 2007. IEEE/RSJ International Conference on, IEEE, pp 1901–1907
23. Krzysztofowicz R, Long D (1990) Fusion of detection probabilities and comparison of multisensor systems. Syst Man Cybern IEEE Trans 20:665–677 (IEEE)
24. Leith DJ, Heidl M, Ringwood JV (2004) Gaussian Process prior models for electrical load forecasting. In: Probabilistic methods applied to power systems, 2004 International Conference on, IEEE, pp 112–117
25. Livingston MA, Sebastian J, Ai Z, Decker JW (2012) Performance measurements for the Microsoft Kinect skeleton. In: Virtual reality short papers and posters (VRW), 2012 IEEE, IEEE, pp 119–120
26. Llinas J, Hall DL, Liggins ME (2009) Handbook of Multisensor data fusion: theory and practice, CRC Press
27. Lu Z, Chen X, Li Q, Zhang X, Zhou P (2014) A hand gesture recognition framework and wearable gesture-based interaction prototype for mobile devices. Hum Mach Systems IEEE Trans 44:293–299
28. Luo RC, Chang CC, Lai CC (2011) Multisensor fusion and integration: theories, applications, and its perspectives. Sens J IEEE 11:3122–3138 (IEEE)
29. MacKay DJ (1998) Introduction to Gaussian processes. NATO ASI Ser F Comput Syst Sci 168:133–166 (Springer Verlag)
30. MacKenzie IS, Ware C (1993) Lag as a determinant of human performance in interactive systems. In: Proceedings of the INTERACT'93 and CHI'93 conference on Human factors in computing systems, ACM, pp 488–493
31. Motion L (2015) Leap motion controller. https://developer.leapmotion.com/. Accessed Apr 2015
32. Murphy RR (1996) Biological and cognitive foundations of intelligent sensor fusion. Syst Man Cybern Part A Syst Hum IEEE Trans 26:42–51 (IEEE)
33. Murray-Smith R, Pearlmutter BA (2005) Transformations of Gaussian process priors. In: Proceedings of the first international conference on deterministic and statistical methods in machine learning,Springer-Verlag, Berlin, Heidelberg, pp 110–123
34. Norrie L, Koelle M, Murray-Smith R, Kranz M (2013) Putting books back on the shelf: Situated interactions with digital book collections on smartphones. In: Proceedings of the 12th international conference on mobile and ubiquitous multimedia MUM '13, ACM, pp 44:1–44:2
35. OpenNI (2014) OpenNI. http://www.openni.org/. Accessed Jan 2014
36. Pavlovych A, Gutwin C (2012) Assessing target acquisition and tracking performance for complex moving targets in the presence of latency and jitter. In: Proceedings of graphics interface 2012 GI '12, Canadian Information Processing Society, pp 109–116
37. Pavlovych A, Stuerzlinger W (2009) The tradeoff between spatial jitter and latency in pointing tasks. In: Proceedings of the 1st ACM SIGCHI symposium on Engineering interactive computing systems, ACM, pp 187–196
38. Rasmussen CE, Williams CKI (2005) Gaussian processes for machine learning (adaptive computation and machine learning), The MIT Press
39. Rohs M, Oulasvirta A (2008) Target acquisition with camera phones when used as magic lenses. In: Proceedings of the SIGCHI conference on human factors in computing systems, ACM, pp 1409–1418
40. Roweis S, Ghahramani Z (1999) A unifying review of linear Gaussian models. Neural Comput 11:305–345 (MIT Press)
41. Schmidt A (2000) Implicit human computer interaction through context. Pers Technol 4:191–199 (Springer)
42. Shneiderman B, Plaisant C (2005) Designing the user interface, edition4 edn. Pearson Addison Wesley, Reading
43. Smith D, Singh S (2006) Approaches to multisensor data fusion in target tracking: a survey. Knowl Data Eng IEEE Trans 18:1696–1710 (IEEE)
44. Solak E, Murray-Smith R, Leithead WE, Leith DJ, Rasmussen CE (2003) Derivative observations in Gaussian Process models of dynamic systems. In: Becker STS, Obermayer K (eds) Advances in neural information processing systems, vol 15. MIT Press, Cambridge, pp 1033–1040
45. Stanton NA, Walker GH et al (2013) Human factors methods: a practical guide for engineering and design. Ashgate Publishing Ltd
46. Strachan S, Murray-Smith R (2009) Bearing-based selection in mobile spatial interaction. Personal Ubiquitous Comput 13:265–280 (Springer-Verlag)
47. Susperregi L, Arruti A, Jauregi E, Sierra B, Martínez-Otzeta JM, Lazkano E, Ansuategui A (2013) Fusing multiple image transformations and a thermal sensor with Kinect to improve person detection ability. Eng Appl Artif Intell 26:1980–1991 (Elsevier)
48. Thrun S (2002) Probabilistic robotics. Commun ACM 45:52–57 (ACM)
49. Titterton D, Weston J (2004) Strapdown inertial navigation technology, vol 17. Peter Peregrinus Ltd
50. Turner RD (2012) Gaussian Processes for state space models and change point detection. PhD thesis, University of Cambridge
51. Vasudevan S (2012) Data fusion with Gaussian processes. Robot Auton Syst 60:1528–1544 (Elsevier)
52. Wang JM, Fleet DJ, Hertzmann A (2008) Gaussian Process dynamical models for human motion. Pattern Anal Mach Intell IEEE Trans 30:283–298 (IEEE)
53. Ware C, Balakrishnan R (1994) Reaching for objects in VR displays: lag and frame rate. ACM Trans Comput Hum Interact (TOCHI) 1:331–356 (ACM)
54. Welch G, Bishop G (1995) An introduction to the Kalman filter, vol 7. University of North Carolina at Chapel Hill, Chapel Hill
55. Welch G, Bishop G (1997) SCAAT: Incremental tracking with incomplete information. In: Proceedings of the 24th annual conference on computer graphics and interactive techniques SIGGRAPH '97. ACM Press/Addison-Wesley Publishing Co, New York, pp 333–344
56. Williamson J (2006) Continuous uncertain interaction. Phd thesis, University of Glasgow
57. Williamson J, Murray-Smith R, Hughes S (2007) Shoogle: excitatory multimodal interaction on mobile devices. In: Proceedings of the SIGCHI conference on Human factors in computing systems CHI '07, pp 121–124
58. Yee K-P (2003) Peephole displays: pen interaction on spatially aware handheld computers. In: Proceedings of the SIGCHI conference on human factors in computing systems. ACM, pp 1–8
59. Zaraki A, Mazzei D, Giuliani M, De Rossi D (2014) Designing and evaluating a social gaze-control system for a humanoid robot. Hum Mach Syst IEEE Trans 44:157–168
60. Zhang Z-Q, Ji L-Y, Huang Z-P, Wu J-K (2012) Adaptive information fusion for human upper limb movement estimation. Syst Man Cybern Part A Syst Hum IEEE Trans 42:1100–1108 (IEEE)