Mechatronics 76 (2021) 102570

Contents lists available at ScienceDirect

Mechatronics

journal homepage: www.elsevier.com/locate/mechatronics

Variable structure Human Intention Estimator with mobility and vision constraints as model selection criteria

Daniel Trombetta, Ghananeel Rotithor, Iman Salehi, Ashwin P. Dani *

Electrical and Computer Engineering, University of Connecticut, Storrs, CT 06269, USA

ARTICLE INFO

Keywords:
Human Intention Estimation
Safe human–robot collaboration/interaction
Multiple model filtering
Fusion

ABSTRACT

In this paper, a novel method for early estimation of a human's action intention is presented. Human intention, modeled as a goal location associated with a hand motion and eye gaze dynamics, is inferred by fusing information from collected hand motion and gaze motion data. The algorithm, called Human Intention Estimator with Variable Structure (HIEVS), uses two variable structure Interacting Multiple Model (VS-IMM) filters in parallel to process the hand motion and gaze data and generate posterior model probabilities associated with a finite set of action models. The posterior model probabilities from each filter are then fused at the end of each iteration and the current intention is estimated as the model, which has the highest fused posterior model probability. Two model set augmentation (MSA) algorithms are presented to select the active models for each VS-IMM during each iteration. For the hand motion filter, an MSA algorithm which computes the human's reachable workspace is used. The MSA algorithm for the gaze filter utilizes the human's visual span to determine the active models. This method allows for accurate early prediction of the human's intention even when the total model set is large. A real world experiment is performed to validate the proposed method.

1. Introduction

Human–Robot (HR) Collaboration has been a growing area of research in recent years [1,2]. The work in this area aims to overcome the constraints which robots encounter in manufacturing, surgical and rehabilitative scenarios. For example, on an assembly line, it is often the case that many different machines, dedicated to single tasks, are required in order to fully assemble a final product. In such cases, the cost of the various machines, installation, and maintenance can easily become burdensome. Additionally, current robots are often limited in the tasks which require a high level of dexterity, whereas humans have better cognitive skills and flexibility to perform dexterous operations. HR Collaboration allows humans and robots to work directly with one another without the need for physical barriers between them. This broadens the span of operations which can be performed by the team of humans and robots in a cost effective manner but also brings the challenges of safety of humans working around robots and efficiency of the team work. To improve the human safety and efficiency, it is required that the robot first be able to interpret the human's action intentions in order to assess its own appropriate actions. The intention estimation can also improve the speed of the current HR collaborative operations.

In [3–5], it is shown that when two humans interact, they infer each others intent in order to safely and effectively collaborate. When humans and robots collaborate, inference of the human's intention improves the overall performance of the task [6–8]. Characteristics of the objects in the workspace [9], human movement [10], heart rate and skin response [11] or information using electromyography (EMG) [12], ultrasound [13] have been used to infer the intention. In [12], a layered hidden Markov model (HMM) is used for intent modeling and estimation using sEMG data. A linear time invariant (LTI) transfer function model is used in [8] for capturing the human response and intent estimation. Intention inference as a goal-reaching motion profile estimation for collaboratively carrying heavy objects is presented in [14]. In [15], a Gaussian Process (GP) is used to predict hand trajectories during an object handover task. Multiple model approach to represent human hand trajectories for learning grasp behaviors using fuzzy logic is presented in [16]. In this paper, a dynamic neural network (DNN) is used to capture the nonlinear effects of the human motion and a point-attractor dynamics is used to model eye gaze to capture point to point converging motion of the gaze.

Recently, the measurement and estimation of 3-dimensional (3D) human eye gaze has been used in human intention schemes. The study presented in [17] suggests a human's gaze is directly related to their intended actions. It is demonstrated in [18] that humans predict action goals by fixating on the end location of an action even before it is

* Corresponding author.
E-mail address: ashwin.dani@uconn.edu (A.P. Dani).

https://doi.org/10.1016/j.mechatronics.2021.102570
Received 23 November 2020; Received in revised form 31 March 2021; Accepted 17 April 2021

reached. In [19], a robot assistant is given instructions correlated to a human intention measure acquired from the human's gaze. In [20], a method to communicate bi-directional navigation intent is developed using augmented reality (AR) and eye-tracking for improved safety in HR interaction. The intention to handover an object is predicted by using key features extracted from the vision and the pose (position + orientation) data in [21]. In [22], gaze information estimated from RGB (Red-Green-Blue) camera images using a convolution neural network (CNN) model is utilized to initialize model probabilities for hand motion tracking using an Interacting Multiple Model (IMM) filter [23]. The methods in [22,24], however, does not utilize gaze information after initialization. The work in [25] presents the human intention algorithm in which both gaze and motion data are utilized in every iteration of a fixed structure IMM (FS-IMM) filter. The method, however, is not computationally efficient when the intention estimator needs to choose from a large number of possible intention models.

To overcome the computational efficiency challenge when a large number of action intention models are present, the work in this paper utilizes variable structure IMM (VS-IMM) filter. The VS-IMM algorithm, presented in [26–28], is similar in structure to a FS-IMM except that at the beginning of each iteration, a model set augmentation (MSA) algorithm selects the most likely subset of the total model set. The filters are only run for models corresponding to the active model set. By constraining the active models in the model set, a large number of total possible intention models can be considered without necessarily increasing the computational burden.

A novel human intention estimation method, called Human Intention Estimator with Variable Structure (HIEVS), is proposed in this paper. The algorithm utilizes two VS-IMM filters [28] running in parallel. One filter produces model probabilities by processing human hand position data while the other simultaneously generates model probabilities by processing human gaze data. Once the posterior model probabilities are made available by each filter they are fused according to the method presented in [25]. Two MSA algorithms are presented to select active model sets for each filter. The MSA algorithms hinge on the concepts of the human's reachable workspace and their peripheral span at each time instance. Using the fused model probabilities, the algorithm predicts the current model which the human is operating under where the model is comprised of a human dynamics which is learned in an initial training phase and a goal location to which the learned dynamics converge. The contributions of the paper are summarized as follows.

- A method to fuse human hand and eye gaze information using VS-IMM filter is presented to estimate the reaching intent when large set of objects are available to reach. The uncertain human hand motion is modeled using a dynamic NN (DNN), which is learned subject to contraction constraints such that the motion profile converges to the object location. The eye gaze motion model is developed using a novel point attractor dynamics.
- A human arm reachable space is computed for selecting active hand motion models. Similarly, human head position tracking and 3D gaze point prediction along with object location is used to compute active models for eye-gaze filter. A fusion equation is developed to fuse the posterior model probabilities of the hand motion and eye-gaze filter.
- Two sets of experiments are conducted to test the HIEVS algorithm for object reaching motions. Both experiments use large number of objects kept on a table that can be reached by the human. For the second set of experiments, the objects are clustered and kept close to each other in a cluster to test the performance of the HIEVS algorithm in the presence of closely kept objects.

Section 2 provides a description of an example problem which is solved using the HIEVS algorithm. The motion models used to represent human hand motion and eye gaze are presented in Section 3, and the method used to learn their respective acceleration functions is presented in Section 4. The novel MSA algorithms used to select the active model sets for each filter are described in Section 5. The HIEVS algorithm is presented in Section 6. In Section 7 experiments are described which show the utility of the HIEVS algorithm in object reaching task, followed by conclusions given in Section 8.

2. Problem formulation and solution approach

Consider a scenario wherein a human and a robot are collaboratively performing an assembly task in a large warehouse environment. The human and the robot are both aware of all N components in the assembly, the location of the components within the warehouse, and how to attach the component to the assembly. This information is often available in a manufacturing warehouse scenario via a Bill of Materials (BoM) and a set of building instructions. In this work, it is assumed that each component is defined by its 3D coordinates within the warehouse, defined as $\mathcal{G} = [g_1, g_2, \dots, g_N]$ and is associated with exactly one building instruction, which is represented as a gaze and hand motion profile, that is taught to the robot via expert demonstrations. The hand motion is described by using a point attached to the palm of the human hand. The goal location, gaze motion, and hand motion tuple is referred to as a model, where $M = [M_1, M_2, \dots, M_N]$ is the entire set of N models. The assembly process is such that components can be attached in many different sequences to achieve the desired result. The human begins to assemble the components in a sequence which he/she sees fit. The robot, who is equipped with 3-dimensional (3D) skeletal tracking data and 3D gaze point measurements of the human, must determine the current step the human is performing so that it may take the appropriate control action for its own motion.

The human's current action is termed as an intent. A block diagram summarizing the HIEVS method, which estimates human intent using the fusion of arm skeletal tracking and human gaze information, is shown in Fig. 1. Multiple reaching motion models are developed by tracking human hand profile data using RGB-D tracking. The acceleration of the human hand profile is modeled using a DNN. The parameters of the DNN are learned such that the position converges to the object location, and velocity and acceleration converge to 0 at the object location. The eye-gaze motion dynamics is developed using a point attractor dynamics, the parameters of which are learned by solving a least squares optimization problem. Two VS-IMM filters are used each for human hand motion and eye gaze motion. For selecting active set of human hand motion VS-IMM filter, reachable space computation of human arm is used which uses 4 joints of the human arm. Human visual span computed using human's head position, predicted gaze point and object position is used to compute the active model set for eye-gaze VS-IMM filter. The posterior model probabilities from two VS-IMM filters are fused to obtain a fused model probability μ_i^F. The reaching intention is then estimated by maximizing the fused model probability. In the following sections, multiple motion modeling, corresponding training methods, active set estimation method and VS-IMM filters are described.

3. Motion models

In this section, human hand motion and human eye-gaze motion models are described in detail.

3.1. Human hand motion

At any given time, the human is assumed to be operating according to one of N models. Let $G = [g_1^T, g_2^T, \dots, g_N^T]^T$ represent the vector of all N goal locations. Then, the ith model M_i is associated with a single goal location g_i. Each model is characterized by the motion of the human hand. The human hand motion associated with the ith model is given by

$$X_H(k+1) = f_i(X_H(k)) + W_H w_1(k) \tag{1}$$

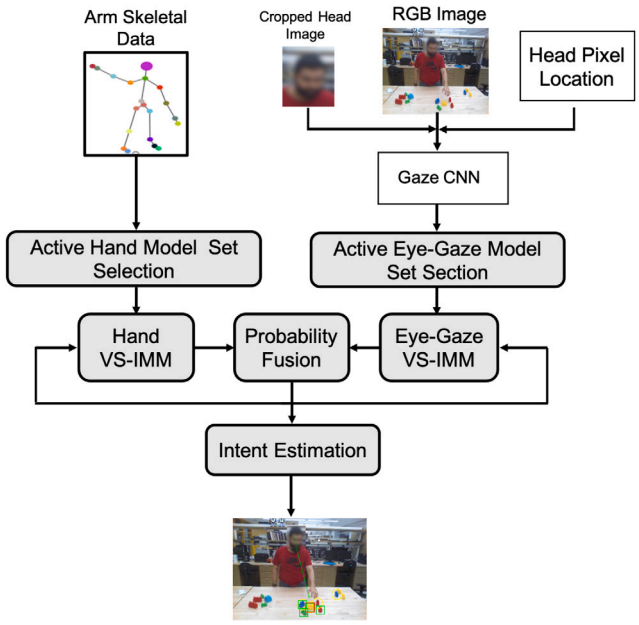


Fig. 1. Block diagram summarizing the components of the HIEVS algorithm.

where $X_H(k) = [x_H^T(k), \dot{x}_H^T(k), \ddot{x}_H^T(k)]^T$, $x_H \in \mathbb{R}^3$ is the 3D position of the human hand, $\dot{x}_H \in \mathbb{R}^3$ and $\ddot{x}_H \in \mathbb{R}^3$ are the corresponding velocities and accelerations, respectively, $W_H = [W_1, W_2, W_3]^T$, $W_1 = \frac{1}{2}T_s^2\mathbb{I}_3$, $W_2 = T_s\mathbb{I}_3$, $W_3 = \mathbb{I}_3$, T_s is the sampling time, and $w_1 \sim \mathcal{N}(0, Q_1)$ is a Gaussian distributed process noise with zero mean and known covariance $Q_1 \in \mathbb{R}^{3\times3}$ that represents the model uncertainty in acceleration update, $f_i : \mathbb{R}^9 \to \mathbb{R}^9$ is given by

$$f_i(X_H(k)) = \begin{bmatrix} \mathbb{I}_3 & T_s\mathbb{I}_3 & \frac{1}{2}T_s^2\mathbb{I}_3 \\ 0 & \mathbb{I}_3 & T_s\mathbb{I}_3 \\ 0 & 0 & 0 \end{bmatrix} X_H(k) + \begin{bmatrix} 0 \\ 0 \\ f_i^H(X_H(k)) \end{bmatrix} \tag{2}$$

where $f_i^H : \mathbb{R}^9 \to \mathbb{R}^3$ is a continuously differentiable function associated with the ith model. Each function f_i^H is approximated by a neural network whose parameters are learned from data collected during the training phase. The training is performed subject to a contraction metric which guarantees even with minimal training data, that predictions made by each f_i^H converge to the ith goal location. A more detailed look at the training method is shown in Section 4. The noisy measurements of the human's hand positions are modeled as

$$z_H(k) = x_H(k) + v_1(k) \tag{3}$$

where $v_1(k) \in \mathbb{R}^{3\times3}$ is a Gaussian distributed measurement noise with zero mean and known covariance $R_1 \in \mathbb{R}^{3\times3}$.

3.2. Eye-gaze motion

Let $X_E(k) = [x_E^T(k), \dot{x}_E^T(k)]^T \in \mathbb{R}^6$ be the state of the eye-gaze motion, where $x_E \in \mathbb{R}^3$ is the 3D gaze-point and $\dot{x}_E \in \mathbb{R}^3$ is the corresponding velocity. The evolution of the human's gaze-point associated with the ith model is given by the discretized point attractor dynamics

$$X_E(k+1) = \begin{bmatrix} \mathbb{I}_3 & T_s\mathbb{I}_3 \\ -K_P T_s\mathbb{I}_3 & (1 - K_D T_s)\mathbb{I}_3 \end{bmatrix} X_E(k) + \begin{bmatrix} \mathbf{0}_3 \\ K_P T_s g_i \end{bmatrix} + \begin{bmatrix} W_1 \\ W_2 \end{bmatrix} w_2(k) \tag{4}$$

where K_P and K_D are the positive scalar gains learned during the training phase, and $w_2 \sim \mathcal{N}(0, Q_2)$ is a Gaussian distributed process noise with zero mean and known covariance $Q_2 \in \mathbb{R}^{3\times3}$ that represents the model uncertainty in the acceleration update. The noisy measurements of the human partner's eye-gaze positions are modeled as

$$z_E(k) = x_E(k) + v_2(k) \tag{5}$$

where $v_2(k) \in \mathbb{R}^{3\times3}$ is a Gaussian distributed measurement noise with zero mean and known covariance $R_2 \in \mathbb{R}^{3\times3}$.

4. Learning hand motion and eye-gaze motion dynamics

In this section, methods used to learn the parameters of the human hand motion and eye-gaze motion dynamics are described.

4.1. Human hand motion model learning

The hand motion dynamics model is learned such that the learned motion model f_i converge to the goal $[g_i^T, \mathbf{0}_3^T, \mathbf{0}_3^T]^T$ regardless of the initial condition. Recall that in this paper, human intention is defined as a motion profile which converges to a single goal location. Thus, it is required that each EKF associated with the ith model makes state predictions which tend toward the ith goal location.

Consider a set of N_D demonstrations $\{\mathcal{D}_i\}_{i=1}^{N_D}$ where $\{X_H(k)\}_{k=0}^{T}$ are recorded from time instances $k = 0$ to time $k = T$. Note that these demonstrations represent reaching motions toward various goal locations when learning DNN approximations of the f_i^H which is a component of f_i. The collected trajectories in $\mathcal{D}_i$ are each translated such that they converge to the origin. Let the translated demonstrations be solutions to the dynamical system governed by the first order differential equation given in (1).

The function $f_i^H(\cdot)$ is modeled using a neural network of the form

$$f_i^H(X_H(k)) = W^T\sigma(U^T s(k)) + \epsilon(s(k)) \tag{6}$$

where $s(k) = [X_H(k), 1]^T \in \mathbb{R}^{10}$ is the input vector to the DNN, $U \in \mathbb{R}^{10\times n_h}$ and $W \in \mathbb{R}^{n_h\times3}$ are the input and output bounded constant weight matrices respectively, $\epsilon(s(k)) \in \mathbb{R}^3$ is the function reconstruction error that goes to zero after the DNN is fully trained, n_h is the number of neurons in the hidden layer of the DNN, $\sigma(U^T s(k)) = [\frac{1}{1+\exp(-(U^T s(k))_1)}, \dots, \frac{1}{1+\exp(-(U^T s(k))_i)}, \dots, \frac{1}{1+\exp(-(U^T s(k))_{n_h})}]^T$ is the vector-sigmoid activation function and $(U^T s(k))_i$ is the ith element of the vector $(U^T s(k))$.

In order to train a contracting DNN, the contraction analysis for discrete time systems is used. The constrained optimization problem to train contracting DNN is given by

$$\{\hat{W}, \hat{U}\} = \arg\min_{W,U}\{E_D + \kappa E_W\} \tag{7}$$

such that

$$\frac{\partial f_i^T}{\partial X_H} M_{k+1} \frac{\partial f_i}{\partial X_H} - M_k \le -\gamma M_k, \qquad M_k \succ 0 \tag{8}$$

where $E_D = \sum_{i=1}^{N_D} \|y_i - a_i\|^2$, $y_i \in \mathbb{R}^{9T\times1}$, y_i represents the target and $a_i = \{X_H(k+1)\}_{k=0}^{T}$ is the output of (2) with f_i^H approximated using a DNN in (6) using ith demonstration, E_W is the sum of the squares of the DNN weights, $\kappa \in \mathbb{R}^+$ is a scalar parameter of regularization, $\gamma \in \mathbb{R}$ is a strictly positive constant, $M_k \in \mathbb{R}^{9\times9}$ represents a uniformly positive definite (PD) contraction metric which is a PD symmetric matrix [29], and the last three rows of the Jacobian $\frac{\partial f_i}{\partial X_H}$ are given by

$$\frac{\partial f_i^H}{\partial X_H} = W^T\frac{\partial\sigma(U^T s)}{\partial X_H} = W^T[\Sigma'(U^T s)]U_x^T \tag{9}$$

where for any $b \in \mathbb{R}^p$, $\Sigma'(b) \in \mathbb{R}^{n_h\times n_h}$ is a diagonal matrix given by

$$\Sigma'(b) = \mathrm{diag}(\sigma(b_1)(1-\sigma(b_1)), \sigma(b_2)(1-\sigma(b_2)), \dots, \sigma(b_p)(1-\sigma(b_p))) \tag{10}$$

and $U_x \in \mathbb{R}^{9\times n_h}$ is a sub-matrix of U formed by taking the first n rows of U.

4.2. Eye-gaze motion model learning

Given a set of $N_{\mathcal{E}}$ goal-converging demonstration trajectories consisting of $\{X_E(k)\}_{k=0}^{T}$ recorded from time instances $k = 0$ to time $k = T$, the scalar positive gains are computed by solving the least squares optimization problem

$$\{\hat{K}_P, \hat{K}_D\} = \arg\min_{K} \sum_{j=1}^{N_{\mathcal{E}}} \sum_{k=0}^{T-1} \|A^j(k, k+1) - Y^j(k)\, K\|^2 \tag{11}$$

where $A^j(k,k+1) = \frac{\dot{x}_E^j(k+1) - \dot{x}_E^j(k)}{T_s}$, $Y^j(k) = \begin{bmatrix} \left(g_j - x_E^j(k)\right) & -\dot{x}_E^j(k) \end{bmatrix}$ and $K = \begin{bmatrix} K_P & K_D \end{bmatrix}^T$. The superscript indicates the iteration over the j^{th} trajectory.

5. Model set augmentation

A key step in the design of VS-IMM filters is the selection of a method to augment the current model set being considered by the VS-IMM. This section presents the nomenclature used in rest of the paper to represent the various model sets and the details of the MSA algorithms used for each filter.

5.1. Definition of model sets

Model sets are defined as follows:

- $M = [M_1, M_2, \ldots, M_N]$ is the complete model set of N possible models. Each model M_i is associated with a single motion profile and goal location g_i where $G = [g_1, g_2, \ldots, g_N]$. Due to the definition of human intention in this work and the one-to-one relationship of models to goal locations, the terms model, intention, and goal location can be used interchangeably in this paper
- $M_a^E(k)$ is the active model set available to the eye-gaze filter at time k, which means that the model probabilities of the models in $M_a^E(k)$ are non-zero.
- $M_r^E(k)$ is the inactive model set reserved by the eye-gaze filter at time k, which means that the model probabilities of the models in $M_r^E(k)$ are zero.
- $M_a^H(k)$ is the active model set available to the hand motion filter at time k, which means that the model probabilities of the models in $M_a^H(k)$ are non-zero.
- $M_r^H(k)$ is the inactive model set reserved by the hand motion filter at time k, which means that the model probabilities of the models in $M_r^H(k)$ are zero.

At any given instance,

$$M_a^E(k) \cap M_r^E(k) = M_a^H(k) \cap M_r^H(k) = \emptyset$$
$$M_a^E(k) \cup M_r^E(k) = M_a^H(k) \cup M_r^H(k) = M$$

In order to utilize the VS-IMM framework, a valid MSA technique must be chosen. In [30], it is stated that a general MSA approach should possess the following properties.

1. It provides a general criterion for model activation and termination. The criterion serves as a general measure of the closeness between the true mode and the candidate models with different structures or parameters.
2. It is computationally feasible. The MSA process can be applied easily with an acceptable computational burden. This property is especially important for models characterized by continuous parameters. It requires that the MSA algorithm should provide a scheme to generate new models from the continuous mode space.
3. It is independent of filters. This requirement allows the MSA algorithm to depend only on the models themselves, and thus can exclude effects of various filters.

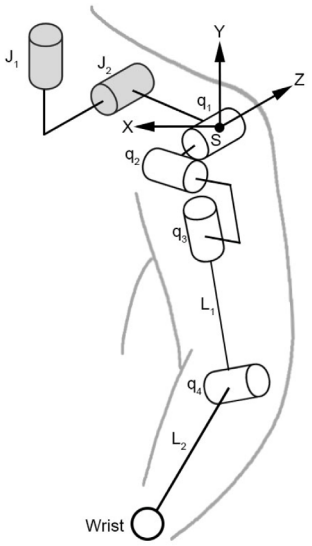


Fig. 2. The simplified human arm model is shown above. Joints J_1 and J_2 are not used in this work. The point S denotes the shoulder position as detected by the Kinect sensor with the positive directions of the Kinect coordinate system attached.

With the properties above in mind, two MSA algorithms are proposed which use constraints on the human's reachable workspace to activate the models M_a^H available to the hand tracking filter and the limitations on human peripheral vision to select the models M_a^E available to the eye-gaze tracking filter.

5.2. Model set augmentation using reachable workspace constraints

The hand motion-tracking VS-IMM filter, which processes the human's 3D wrist position data acquired through skeletal tracking, generates probabilities that any given model in the set M is the correct model at time k based on the measured 3D wrist position at time k. However, it is unlikely that models corresponding to goal locations which are not within the human's reachable workspace are the true model. Thus, models whose goal locations lie outside of the reachable workspace need not be considered in the filter.

In [31,32], a method of evaluating a human's reachable space is presented which uses a six degree-of-freedom (DoF) model of the human arm, comprised of six revolute joints, and joint limits of a healthy subject to determine the region which is reachable by the human's wrist relative to their shoulder. By evaluating this region before the onset of the intention inference process and assuming it to be constant relative to the shoulder joint, which is also tracked during the process by the skeletal tracker, the goal locations which lie within the reachable region can be determined.

In order to estimate the reachable workspace, denoted $\mathcal{R}_{ws}$, the six DoF human arm model is simplified further to a four DoF model by eliminating the two DoF that model flexion–extension and abduction–adduction in the inner shoulder joint and produce minimal effects in wrist position. The simplified model is shown in Fig. 2 with the eliminated joints shaded in gray. The elbow joint (q_4) is held constant at 0 degrees while the other three joints (q_1, q_2, q_3) are sampled at 100 evenly spaced points between their joint limits in order to obtain the outer-most possible wrist positions. Due to the anatomical properties of the arm, bones and muscles, some joint limits are dependent on the positions of the other joint angles. The joint limits are given as

$$\begin{aligned} q_1 &\in [-9^\circ, 160^\circ] \\ q_2 &\in \left[\left(-43 + \frac{q_1}{3}\right)^\circ, \left(153 - \frac{q_1}{6}\right)^\circ\right] \\ q_3 &\in \left[\left(-90 + \frac{7q_1}{9} - \frac{q_2}{9} + \frac{2q_1 q_2}{810}\right)^\circ, \right. \end{aligned} \tag{12}$$

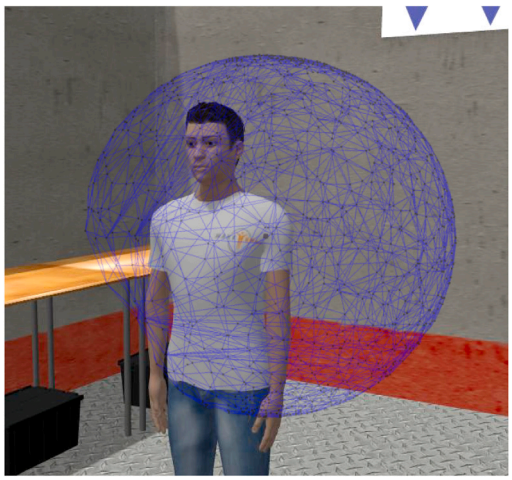


Fig. 3. The region $\mathcal{R}_{ws}$ calculated using the method described in Section 5.2 overlain on a simulated human worker in a warehouse scene.

$$\left(60 + \frac{4q_1}{9} - \frac{5q_2}{9} + \frac{5q_1q_2}{810}\right)^{\circ}\Bigg]$$

The region $\mathcal{R}_{ws}$, shown in Fig. 3 is the volume bounded by the resulting wrist positions. In each iteration, the active model set available to the hand-tracking filter is then chosen to be

$$M_a^H = \mathcal{R}_{ws} \cap G = \mathcal{R}_{ws} \cap M \tag{13}$$

5.3. Model set augmentation using human visual span

In the vision literature, the human visual span or the peripheral span, refers to the region of the visual field from which one can extract information during an eye fixation [33]. Similar to the method described in Section 5.2, if a goal location does not lie within the visual span, it is unlikely to be the true goal location. In [33], a series of experiments are performed to measure the human visual span during an object search in real-world scenes. The results show that the average visual span during an object search is a cone whose aperture has a radius of about 8 degrees.

Let $\overline{vs}$ be the vector from the position of the human's eyes, estimated as $X_{head} - X_g$, where X_{head} is the position of the head detected by the Kinect skeletal tracking in Kinect reference frame and X_g is the gaze point computed using the deep network in [34]. Then, for the eye-gaze filter, the models that are chosen to be active at time k are those which fall within the region $\mathcal{R}_{vs}$ defined with respect to F_K as the volume of a cone centered about $\overline{vs}$ with a radius of 8 degrees. That is

$$M_a^E = \mathcal{R}_{vs} \cap G = \mathcal{R}_{vs} \cap M \tag{14}$$

5.4. Addition and removal of active models

Let μ_i^F be a fused model probability obtained from the posterior model probabilities of human hand motion filter and eye-gaze filter. When a model which was active in the previous iteration becomes inactive, its corresponding model probability is set to be zero, and the filter corresponding to the model is made inactive. That is, if model M_i with non-zero fused model probability μ_i^F was in the set M_a^H at time $k-1$, but at time k the MSA algorithm has determined that M_i should be removed from M_a^H, then M_i is added to the set M_r^H, $\mu_i^H = 0$, and the ith filter does not run on the kth iteration. The same logic holds true for the eye-gaze filter.

On the other hand, when a model which was previously inactive to both filters and thus had a model probability $\mu_i^F = 0$ at time $k-1$ becomes active at time k, its fused model probability is initialized to a small threshold value π_{th} and its associated filter is made active for the kth iteration. If ω models which were previously inactive become active at time k, then their associated model probabilities are initialized equally as

$$\mu_i^F = \frac{\pi_{th}}{\omega} \tag{15}$$

It is important to reiterate that the initialization described above needs to be performed if and only if $M_i \in M_r^H$ and $M_i \in M_r^E$ during the previous iteration. Otherwise, the associated fused model probability will be non-zero. Once a new model is added to M_a^E or M_a^H the model probabilities of all the active models are renormalized.

6. VS-IMM human intention inference algorithm

In this section, the Human Intention Estimator with Variable Structure (HIEVS) algorithm is presented. The algorithm consists of two VS-IMM filters running in parallel. One filter processes eye-gaze data in order to produce an estimate of the eye-gaze point $\hat{x}_E(k)$ and the set of posterior model probabilities conditioned on gaze point measurements associated with each model $\mu_i^E(k)$. The other filter processes hand motion data in order to produce an estimate of the hand position $\hat{x}_H(k)$ and the set of posterior model probabilities conditioned on the hand position measurements associated with each model $\mu_i^H(k)$. The initial state and covariance for each filter, i.e. $\hat{x}_H(0|0)$, $P_H(0|0)$ and $\hat{x}_E(0|0)$, $P_E(0|0)$, are acquired using the two-point differencing method. On the first iteration for each filter, the prior model probabilities $\mu_j^F(0)$ are initialized to be uniform across all models in M. Subsequent model probabilities $\mu_j^F(k)$ are acquired from the fusion equations defined at the end of this section.

6.1. Human hand motion filter

The hand position VS-IMM filter is described below.

Interaction/mixing: At the beginning of each iteration, the initial conditions (state estimate $\hat{x}_H^{0j}(k-1|k-1)$ and covariance $\hat{P}_H^{0j}(k-1|k-1)$), where superscript 0 denotes initial condition, j denotes the number of the filter, at time k, are adjusted by mixing the filter outputs from the previous iteration (time instant $k-1$) in the following way

$$\hat{x}_H^{0j}(k-1|k-1) = \sum_{i=1}^{N} \hat{x}_H^i(k-1|k-1) \times \mu_{i|j}^F(k-1|k-1), \qquad j = 1, \ldots, N \tag{16}$$

$$\hat{P}_H^{0j}(k-1|k-1) = \sum_{i=1}^{N} \mu_{i|j}^F(k-1|k-1)\hat{P}_H^i(k-1|k-1) + [\hat{x}_H^i(k-1|k-1) - \hat{x}_H^{0j}(k-1|k-1)] \times [\hat{x}_H^i(k-1|k-1) - \hat{x}_H^{0j}(k-1|k-1)]^T, \quad j = 1, \ldots, N \tag{17}$$

where $\hat{x}_H^i(k-1|k-1)$, $\hat{P}_H^i(k-1|k-1)$ are the state estimate and its covariance, respectively, corresponding to model M_j at time $k-1$ and $\mu_{i|j}^F(k-1|k-1)$ are the mixing probabilities given by

$$\mu_{i|j}^F(k-1|k-1) = \frac{\Pi_{ij}\mu_i^F(k-1)}{\bar{c}_j}, \quad i, j = 1, 2, \ldots, N \tag{18}$$

where $\Pi_{ij} = p(M(k) = M_j | M(k-1) = M_i)$ is the model transition or jump probability and $\mu_i^F(k-1) = p(M_i | Z_H^{1:k-1}, Z_E^{1:k-1})$ is the fused probability of ith model M_i being the right model at time $k-1$ and $\bar{c}_j = \sum_{i=1}^{N} \Pi_{ij}\mu_i^F(k-1)$ are the normalizing constants.

Model matched filtering: Once the initial conditions $\hat{x}_H^{0j}(k-1|k-1)$ and $\hat{P}_H^{0j}(k-1|k-1)$ are available for each filter, the state estimate and its covariance for each model are computed using the EKFs matched to the models. Along with the state estimates and the corresponding covariances, the likelihood functions $\Lambda_j^H(k)$ are computed using the mixed initial condition (16) and the corresponding covariance (17).

The likelihood $\Lambda_j^H(k)$, a Gaussian distribution with the predicted measurement as the mean and the covariance equal to the innovation covariance, is given by

$$\Lambda_j^H(k) = p(z_H(k)|M_j(k), Z_H^{1:k-1})$$
$$\Lambda_j^H(k) = \mathcal{N}(z_H(k); \hat{z}_H^j(k|k-1; \hat{x}_H^{0j}(k-1|k-1)), S_H^j(k; \hat{P}_H^{0j}(k-1|k-1))), \ j = 1, \dots, N \quad (19)$$

where $S_H^j(k; P_H^{0j}(k-1|k-1))$ is the innovation covariance and $\hat{z}_H^j(k|k-1; \hat{x}_H^{0j}(k-1|k-1))$ is the jth filter's predicted measurement at time t.

Model probability update: After the likelihood functions of the models $\Lambda_j^H(k)$ are available, the model posterior probabilities $\mu_j^H(k)$ are calculated as follows

$$\mu_j^H(k) = P(g_j|Z_H^{1:k}) = P(M_j(k)|Z_H^{1:k})$$
$$\mu_j^H(k) = p(z_H(k)|M_j(k), Z_H^{1:k-1})P(M_j(k)|Z_H^{1:k-1})$$
$$\mu_j^H(k) = \frac{\Lambda_j^H(k)\bar{c}_j}{\sum_{i=1}^N \Lambda_i^H(k)\bar{c}_i}, \qquad j = 1, 2, \dots, N \quad (20)$$

6.2. Eye-gaze filter

The eye-gaze VS-IMM filter has a similar form to the hand motion filter described in Section 6.1.

Interaction/mixing: At the beginning of each iteration, the initial conditions of the eye-gaze filter (state estimate $\hat{x}_E^{0j}(k-1|k-1)$ and covariance $\hat{P}_E^{0j}(k-1|k-1)$) are adjusted by mixing the filter outputs from the previous iteration according to

$$\hat{x}_E^{0j}(k-1|k-1) = \sum_{i=1}^N \hat{x}_E^i(k-1|k-1) \times \mu_{i|j}^F(k-1|k-1), j = 1, \dots, N \quad (21)$$

$$\hat{P}_E^{0j}(k-1|k-1) = \sum_{i=1}^N \mu_{i|j}^F(k-1|k-1)\hat{P}_E^i(k-1|k-1) + [\hat{x}_E^i(k-1|k-1) - \hat{x}_E^{0j}(k-1|k-1)] \times [\hat{x}_E^i(k-1|k-1) - \hat{x}_E^{0j}(k-1|k-1)]^T \quad j = 1, \dots, N \quad (22)$$

Model matched filtering: Once the initial conditions $\hat{x}_E^{0j}(k-1|k-1)$ and $\hat{P}_E^{0j}(k-1|k-1)$ are available for each filter, the state estimate and its covariance for each model are computed using the KFs matched to the models. Along with the state estimates and the corresponding covariances, the likelihood functions $\Lambda_j^E(k)$ are computed using the mixed initial condition (21) and the corresponding covariance (22). The likelihood $\Lambda_j^E(k)$ is given by

$$\Lambda_j^E(k) = p(z_E(k)|M_j(k), Z_E^{1:k-1})$$
$$\Lambda_j^E(k) = \mathcal{N}(z_E(k); \hat{z}_E^j(k|k-1; \hat{x}_E^{0j}(k-1|k-1)), S_E^j(k; \hat{P}_E^{0j}(k-1|k-1))), \ j = 1, \dots, N \quad (23)$$

where $S_E^j(k; P_E^{0j}(k-1|k-1))$ is the innovation covariance and $\hat{z}_E^j(k|k-1; \hat{x}_E^{0j}(k-1|k-1))$ is the jth filter's predicted measurement at time t.

Model probability update: After the likelihood functions of the models $\Lambda_j^E(k)$ are available, the model posterior probabilities $\mu_j^E(k)$ are calculated as follows

$$\mu_j^E(k) = P(g_j|Z_E^{1:k}) = P(M_j(k)|Z_E^{1:k})$$
$$\mu_j^E(k) = p(z_E(k)|M_j(k), Z_E^{1:k-1})P(M_j(k)|Z_E^{1:k-1})$$
$$\mu_j^E(k) = \frac{\Lambda_j^E(k)\bar{c}_j}{\sum_{i=1}^N \Lambda_i^E(k)\bar{c}_i}, \qquad j = 1, 2, \dots, N \quad (24)$$

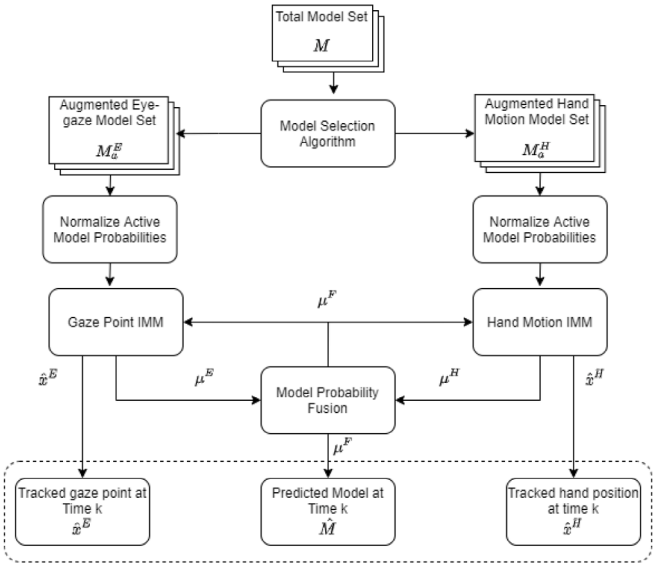


Fig. 4. Detailed block diagram of VS-IMM algorithm used for the HIEVS method.

6.3. Determination of human intention

Once posterior model probabilities are available from both filters, they are fused using

$$\mu_j^F(k) = \alpha e^{-\beta T_t}\mu_j^E(k) + (1 - \alpha e^{-\beta T_t})\mu_j^H(k) \quad (25)$$

The goal location estimate $\hat{g}(k)$ is then given by

$$\hat{g}(k) = \arg \max_{g \in M_a^H \cup M_a^E} \mu_j^F(k) \quad (26)$$

The optimization problem in (26) is solved by choosing the location $g_i \in M_a^H \cup M_a^E$ corresponding to the model M_i with the highest model probability $\mu_i^F(k)$ at time k. The goal location search in (26) is performed over the union of sets M_a^H and M_a^E to consider cases when these sets have different cardinality or an empty intersection. Fig. 4 summarizes the gaze and motion fusion algorithm in the form of a block diagram.

7. Experimental evaluation

7.1. Experimental setup

An experiment which simulates an assembly task in a large warehouse setting is designed to verify the utility the proposed method. A Microsoft Kinect sensor is placed such that a large workstation is fully visible. Within the workstation, tools corresponding to $N = 18$ for experiment 1 and $N = 13$ for experiment 2 models are placed arbitrarily but at coordinates which are known to the algorithm a priori with respect to the Kinect reference frame. During this process, the Kinect sensor records 3D skeletal tracking data and RGB images. The CNN architecture, presented in [34], is used to predict the 3D gaze point of the human. The input to the neural network is an RGB image obtained from the Kinect, the pixel location of the head and a cropped head image obtained from the 3D skeletal tracking. The cropped head image is obtained using the Viola–Jones face detector. The CNN consists of a head condition branch and a scene branch, each consisting of a ResNet-50 CNN followed by an additional residual pooling layer. The output of these two branches is merged using a two-layer encode module and is input to a recurrent attention prediction module which consists of a Convolutional-LSTM network followed by four deconvolution layers to predict a full-sized heatmap. The final 3D gaze point of the human is then computed using this heatmap. The CNN is trained on the

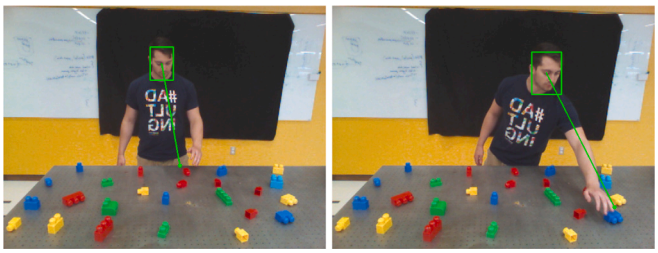


Fig. 5. Two RGB images taken from the Kinect sensor corresponding to frames 10 and 105. The images have been overlain with bounding boxes around the face and a vector from the face to the predicted gaze point.

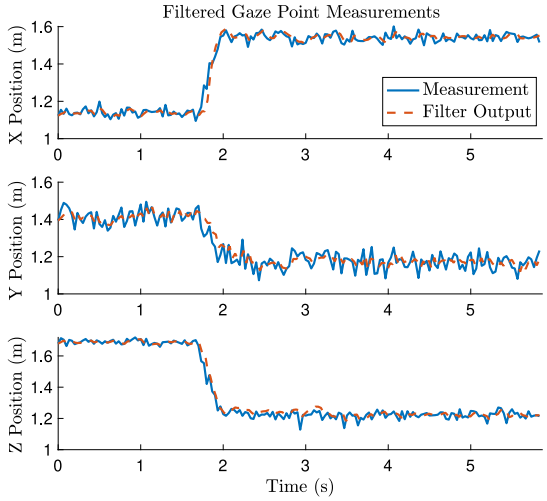


Fig. 6. The human gaze point tracked by the HIEVS algorithm is shown in dotted red. The measurements, in solid blue, are acquired by passing the RGB images collect by the Kinect sensor through a deep network for gaze point prediction.

VideoAttentionTarget dataset created specifically by the authors for the task of 3D gaze point prediction in video sequences. An example of the output of the network is shown in Fig. 5. The algorithm is run using MATLAB 2020b on an Intel i7 processor computer with 8 cores and 32 GB RAM. Two NVIDIA RTX 2080 GPUs are used for the inference of deep learning algorithm.

The DNN model in Section 4.1 uses a single-layer contracting NN with 15 neurons. $N_D = 5$ are used to train the DNN model with each trajectory containing 500 data points sampled at 30 Hz. The data is collected using Kinect sensor tracking human skeleton. The accuracy of the learned DNN model was comparable to the DNN model from our prior work presented in [22]. For the eye-gaze model in Section 4.2, $N_{\mathcal{E}} = 3$ trajectories are used to train the model containing 500 data points sampled at 30 Hz. The 3D gaze point generated by the CNN is used for training the eye-gaze model.

Since the algorithm uses IMM filters, disturbances like process noise and measurement noise can be accounted for by selecting appropriate state and measurement covariance matrices. In the formulation, the process noise can be due to human hand and eye-gaze model learning errors and the measurement noise is due to the uncertainties from the CNN gaze-point predictions and Kinect skeletal tracking.

7.2. Experiment 1

In this experiment, in order to model the assembly of components, the human chooses and reaches for any two objects which are not adjacent to one another in sequence. The results of experiment 1 show that the HIEVS algorithm successfully tracks hand and gaze point motion and can correctly predict the human's intention in both stages

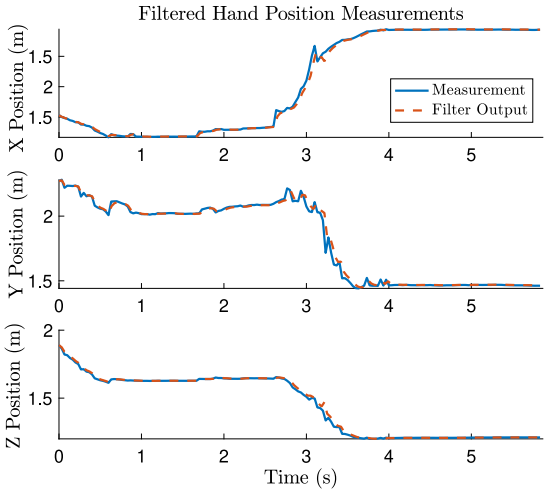


Fig. 7. Human hand motion tracked by the HIEVS algorithm is shown in dotted red. The hand motion data is acquired via the Kinect sensor's skeletal tracking feature.

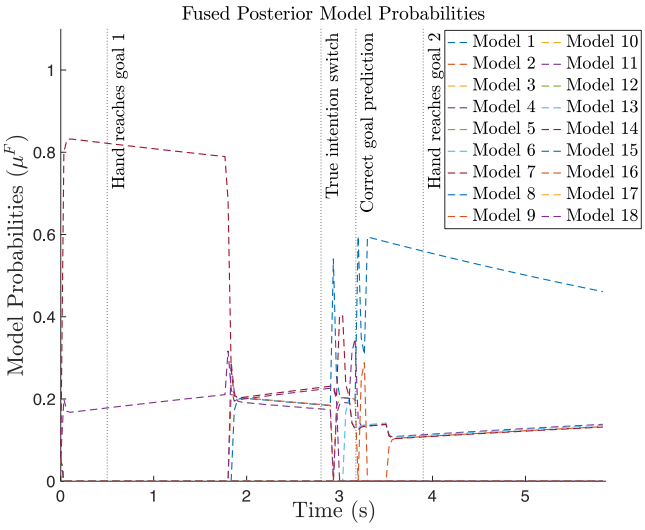


Fig. 8. The evolution of the fused model probabilities associated with the N models. The vertical dotted black line denotes the change in true intention.

of the sequence. The gaze point and hand motion tracked by HIEVS are shown in Figs. 6 and 7, respectively. Fig. 8 shows the evolution of the model probabilities matched to each of the $N = 18$ goal locations. It can be seen in Fig. 6 that the subject's gaze is fixated on the first goal location, corresponding to M_7, from time $t = 0$ s until about $t = 1.8$ s at which point it begins to shift to the second goal location. The saccade to the second goal, M_1, takes about 0.2 s and the gaze remains fixated on this point for the remainder of the trail. Fig. 7 shows that the hand does not start moving toward M_1 until $t = 2.6$ s, nearly a full second after the gaze has shifted. This occurrence is accounted for in the fusion of the posterior model probabilities in (25) by the parameters α and β which were set to be 0.5 and 1, respectively. This means that, at first, both μ^E and μ^H are weighed equally, but as time goes on, μ^H begins to hold more weight. This is observed in Fig. 8 where the dotted red line dips. New models are activated at $t = 1.8$ s because the gaze point shifts. However, because the hand stays in the same location, and its weight is increasing over time, the associated model probability continues increasing. When the hand location shifts around 2.8 s, the algorithm quickly recognizes a change in intention, predicting M_1 to the most likely model at about 3.2 s, which is 0.7 s before the hand reaches the associated goal location, g_1.

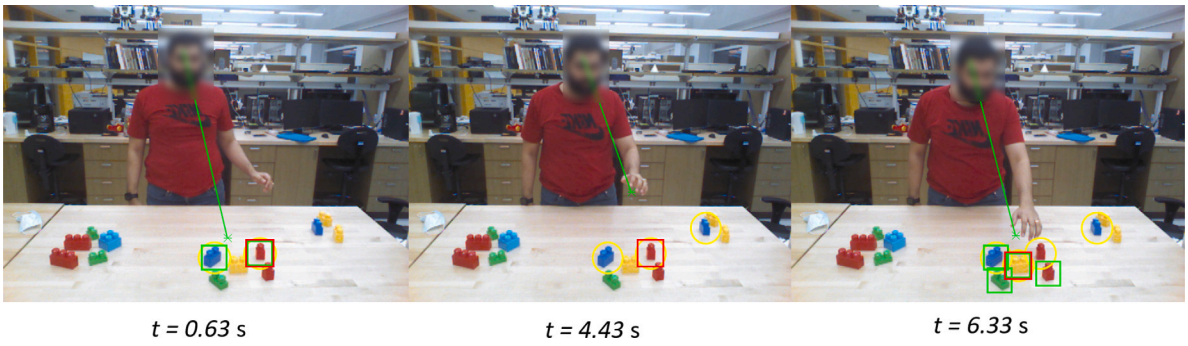


Fig. 9. Sequence of images showing the progression of the estimated gaze point by the deep learning method and the estimated human intent by the HIEVS algorithm. The estimated human intent is marked using the red bounding box, the gaze point is marked by the green * marker, the active set of hand models M_a^H are marked by yellow circles and the active set of gaze models M_a^E are marked by the green bounding boxes. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

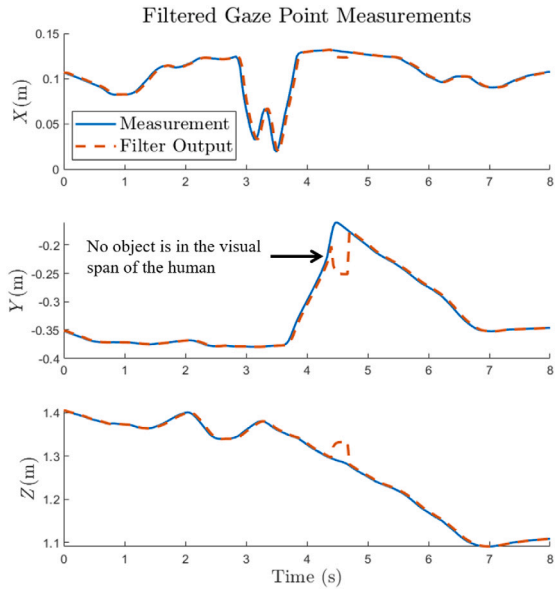


Fig. 10. The human gaze point tracked by the HIEVS algorithm is shown in dotted red. The measurements, in solid blue, are acquired by passing the RGB images collect by the Kinect sensor through a deep network for gaze point prediction. At $t = 4.43$ s the set M_a^E is empty and hence the gaze IMM is not used.

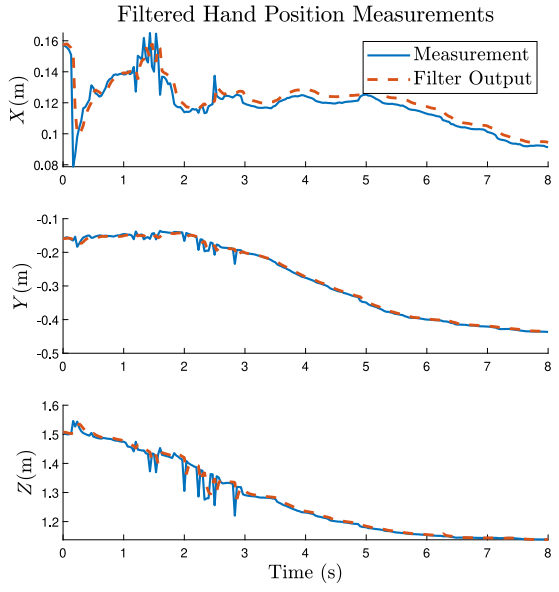


Fig. 11. Human hand motion tracked by the HIEVS algorithm is shown in dotted red. The hand motion data is acquired via the Kinect sensor's skeletal tracking feature.

7.3. Experiment 2

The objective of this experiment is to test the algorithm's performance when certain objects are placed close to each other. For this experiment, $N = 13$ objects are placed in three separate clusters. The objects in each cluster are placed very close to one another with the closest distance as small as 7.5 cm. A test trajectory of human hand reaching motion is used to test the intention estimation using HIEVS algorithm for a ground truth goal reaching motion of hand reaching object number 8 as shown in Fig. 9 at $t = 6.33$ s. It is important to note that the objects are modeled by a point and the distance between any two objects is the distance between these two points. The values of the parameters α and β used for fusing posterior model probabilities are set to 0.7 and 0.2 respectively. Fig. 9 shows the progression of the estimated human intent using HIEVS algorithm. The gaze point and hand motion tracked by HIEVS are shown in Figs. 10 and 11, respectively. At $t = 4.43$ s the deep learning algorithm fails to detect the human gaze correctly which leads to the active model set M_a^E being empty. Since the active model set M_a^E is empty, the prediction of $\hat{g}(k)$ in (26) is obtained from the set M_a^H. This can be visualized in the gaze IMM filter output shown in Fig. 10. The true intention of the human is correctly estimated by the HIEVS algorithm at $t = 6.33$ s as shown in Fig. 9 for a trajectory of length 8 s. Since the objects are very closely placed, the fused posterior model probabilities for objects 7 (blue block), 8 (yellow block) and 9 (red block with yellow circle) are comparable. At $t = 6.33$, the fused posterior model probability value for object 8 is 0.26, while for objects 7 and 9 the fused probabilities are 0.25 and 0.20, respectively. The model with highest posterior model probability is selected as the reaching intention, i.e., model for object 8. At $t = 8$ s, the objects 7, 8, 9 have fused probability values 0.25, 0.26, 0.22, respectively, as shown in Fig. 12. The algorithm correctly estimates the true intention when the human hand is sufficiently close to the true intention.

8. Conclusion

This paper proposes a novel framework for human intention estimation. The algorithm presented, called Human Intention Estimator with Variable Structure (HIEVS), uses two VS-IMM filters running in parallel, one which processes human hand motion data and another which acts on human gaze-point data, and fuses the posterior model probabilities generated by each filter in order to determine a human's action intention. Dynamical models of the human's hand motion and

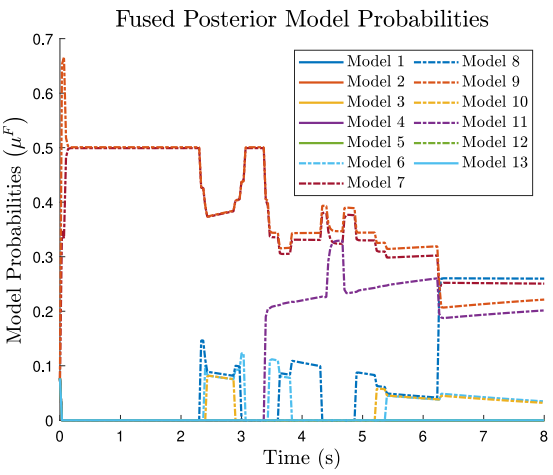


Fig. 12. The evolution of the fused model probabilities associated with the $N = 13$ models with model 8 being the true intention.

eye-gaze motion are learned in an initial training phase. Novel MSA algorithms are used for active model set selection. The MSA algorithm for the hand motion filter uses reachable workspace computation of the human arm to compute reachable active sets. Likewise, the MSA algorithm for the eye-gaze filter uses visual span to compute active model set. Two experiments are conducted which show that the HIEVS algorithm is capable of estimating predictions of human intention when there are large number of objects to select from and when the objects are placed very close to each other. The experiments also reveal that the HIEVS algorithm can predict correct intention when the human user changes the eye-gaze for a second. In future, applicability of the method when the object are moving will be studied.

CRediT authorship contribution statement

Daniel Trombetta: Conceptualization, Methodology, Formal analysis, Software, Writing - original draft. **Ghananeel Rotithor:** Formal analysis, Investigation, Data curation. **Iman Salehi:** Visualization, Investigation. **Ashwin P. Dani:** Conceptualization, Formal analysis, Validation, Writing - review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

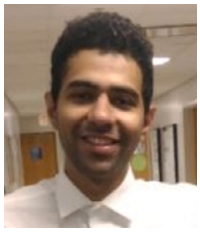
Acknowledgments

This work was supported in part by Space Technology Research Institutes, USA grant (number 80NSSC19K1076) from NASA Space Technology Research Grants Program, USA and in part by the U.S. Department of Energy's Office of Energy Efficiency and Renewable Energy (EERE) under the Advanced Manufacturing Office Award Number DE-EE0007613.

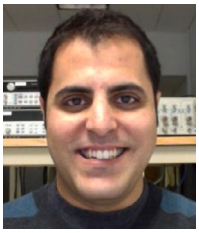
References

[1] Villani V, Pini F, Leali F, Secchi C. Survey on human—robot collaboration in industrial settings: Safety, intuitive interfaces and applications. Mechatronics 2018;55:248–66.
[2] Modares H, Ranatunga I, AlQaudi B, Lewis FL, Popa DO. Intelligent human–robot interaction systems using reinforcement learning and neural networks. In: Trends in Control and Decision-Making for Human–Robot Collaboration Systems. 2017, p. 153–76.
[3] Baldwin DA, Baird JA. Discerning intentions in dynamic human action. Trends Cogn Sci 2001;5(4):171–8.
[4] Cremer S, Das SK, Wijayasinghe IB, Popa DO, Lewis FL. Model-free online neuroadaptive controller with intent estimation for physical human–robot interaction. IEEE Trans Robot 2019;36(1):240–53.
[5] Simon MA. Understanding human action: Social explanation and the vision of social science. SUNY Press; 1982.
[6] Li Y, Ge S. Human-robot collaboration based on motion intention estimation. IEEE/ASME Trans Mechatronics 2014;19(3):1007–14.
[7] Liu C, Hamrick JB, Fisac JF, Dragan AD, Hedrick JK, Sastry SS, Griffiths TL. Goal inference improves objective and perceived performance in human-robot collaboration. In: International Conference on Autonomous Agents & Multiagent Systems. 2016, p. 940–8.
[8] Warrier RB, Devasia S. Inferring intent for novice human-in-the-loop iterative learning control. IEEE Trans Control Syst Technol 2016;25(5):1698–710.
[9] Koppula HS, Gupta R, Saxena A. Learning human activities and object affordances from rgb-d videos. Int J Robot Res 2013;32(8):951–70.
[10] Mainprice J, Hayne R, Berenson D. Predicting human reaching motion in collaborative tasks using inverse optimal control and iterative re-planning. In: IEEE International Conference on Robotics and Automation. 2015, p. 885–92.
[11] Kulic D, Croft EA. Affective state estimation for human–robot interaction. IEEE Trans Robot 2007;23(5):991–1000.
[12] Razin YS, Pluckter K, Ueda J, Feigh K. Predicting task intent from surface electromyography using layered hidden Markov models. IEEE Robot Autom Lett 2017;2(2):1180–5.
[13] Zhang Q, Kim K, Sharma N. Prediction of ankle dorsiflexion moment by combined ultrasound sonography and electromyography. IEEE Trans Neural Syst Rehabil Eng 2019;28(1):318–27.
[14] Ravichandar HC, Trombetta D, Dani AP. Human intention-driven learning control for trajectory synchronization in human-robot collaborative tasks. IFAC-PapersOnLine 2019;51(34):1–7.
[15] Lang M, Endo S, Dunkley O, Hirche S. Object handover prediction using gaussian processes clustered with trajectory classification. 2017, arXiv preprint arXiv:1707.02745.
[16] Palm R, Iliev B. Learning of grasp behaviors for an artificial hand by time clustering and takagi-sugeno modeling. In: 2006 IEEE International Conference on Fuzzy Systems. 2006, p. 291–8.
[17] Yarbus AL. Eye Movements and Vision. Springer; 1967.
[18] Flanagan JR, Johansson RS. Action plans used in action observation. Nature 2003;424(6950):769–71.
[19] Sakita K, Ogawara K, Murakami S, Kawamura K, Ikeuchi K. Flexible cooperation between human and robot by interpreting human intention from gaze information. In: IEEE/RSJ International Conference on Intelligent Robots and Systems. 2004, p. 846–51.
[20] Chadalavada RT, Andreasson H, Schindler M, Palm R, Lilienthal AJ. Bi-directional navigation intent communication using spatial augmented reality and eye-tracking glasses for improved safety in human–robot interaction. Robot Comput-Integr Manuf 2020;61:101830.
[21] Strabala KW, Lee MK, Dragan AD, Forlizzi JL, Srinivasa S, Cakmak M, Micelli V. Towards seamless human-robot handovers. J Human-Robot Interact 2013;2(1):112–32.
[22] Ravichandar HC, Kumar A, Dani A. Gaze and motion information fusion for human intention inference. Int J Intell Robot Appl 2018;2(2):136–48.
[23] Bar-Shalom Y, Li XR, Kirubarajan T. Estimation with Applications to Tracking and Navigation. John Wiley and Sons; 2001.
[24] Strabala K, Lee MK, Dragan A, Forlizzi J, Srinivasa S. Learning the communication of intent prior to physical collaboration. In: IEEE International Symposium on Robot and Human Interactive Communication. 2012.
[25] Trombetta D, Rotithor GS, Salehi I, Dani AP. Human intention estimation using fusion of pupil and hand motion. In: IFAC World Congress. 2020.
[26] Li X-R, Bar-Shalom Y. Multiple-model estimation with variable structure. IEEE Trans Automat Control 1996;41(4):478–93.
[27] Li XR. Multiple-model estimation with variable structure. II. Model-set adaptation. IEEE Trans Automat Control 2000;45(11):2047–60.
[28] Li XR, Zwi X, Zwang Y. Multiple-model estimation with variable structure. III. Model-group switching algorithm. IEEE Trans Aerosp Electron Syst 1999;35(1):225–41.
[29] Lohmiller W, Slotine J-JE. On contraction analysis for nonlinear systems. Automatica 1998;34(6):683–96.
[30] Lan J, Li XR, Mu C. Best model augmentation for variable-structure multiple-model estimation. IEEE Trans Aerosp Electron Syst 2011;47(3):2008–25.
[31] Lenarcic J, Umek A. Simple model of human arm reachable workspace. IEEE Trans Syst Man Cybern 1994;24(8):1239–46.
[32] Kurillo G, Chen A, Bajcsy R, Han JJ. Evaluation of upper extremity reachable workspace using kinect camera. Technol Health Care 2013;21(6):641–56.
[33] Nuthmann A. On the visual span during object search in real-world scenes. Vis Cogn 2013;21(7):803–37.
[34] Chong E, Wang Y, Ruiz N, Rehg JM. Detecting Attended Visual Targets in Video, In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020: p. 5396–5406.

Daniel Trombetta received his B.S. and M.S. in Electrical and Computer Engineering from the University of Connecticut in 2018 and 2020, respectively. He is currently working as an Electrical Engineer at Dynetics. His research interests include machine learning, human–robot collaboration, and estimator design.



Ghananeel Rotithor received the M.S. degree in Biomedical engineering from the University of Florida, Gainesville, FL in 2017. He is currently pursuing his Ph.D. degree from the Electrical and Computer Engineering department at the University of Connecticut, CT. His research interests include deep learning for control, robotics, vision-based estimation and control.



Iman Salehi received his M.S. degree from the Department of Electrical and Computer Engineering at the University of Hartford in 2015. He is currently working toward his Ph.D. in the Electrical and Computer Engineering department at the University of Connecticut, Storrs, CT. His research interests include learning for control, human–robot interaction, and system identification.

Ashwin Dani (SM'18, M'11) received the M.S. and Ph.D. degrees from the University of Florida, Gainesville, FL. He was a Post-Doctoral Research Associate at the University of Illinois, Urbana–Champaign, IL. He is currently an Associate Professor at the University of Connecticut, Storrs, CT. He has authored over 50 technical papers and 4 book chapters. His current research interests include nonlinear estimation and control, machine learning for control, human–robot collaboration, vision-based control and autonomous navigation.