Noname manuscript No.
(will be inserted by the editor)

Gaze and Motion Information Fusion for Human Intention Inference

Harish chaandar Ravichandar · Avnish Kumar · Ashwin Dani

Received: date / Accepted: date

Abstract An algorithm, named gaze-based multiple model intention estimator (G-MMIE), is presented for early prediction of the goal location (intention) of human reaching actions. The trajectories of the arm motion for reaching tasks are modeled by using an autonomous dynamical system with contracting behavior towards the goal location. To represent the dynamics of human arm reaching motion, a neural network (NN) is used. The parameters of the NN are learned under constraints derived based on contraction analysis. The constraints ensure that the trajectories of the dynamical system converge to a single equilibrium point. In order to use the motion model learned from a few demonstrations in new scenarios with multiple candidate goal locations, an interacting multiple-model (IMM) framework is used. For a given reaching motion, multiple models are obtained by translating the equilibrium point of the contracting system to different known candidate locations. Hence, each model corresponds to the reaching motion that ends at the respective candidate location. Further, since humans tend to look toward the location they are reaching for, prior probabilities of the goal locations are calculated based on the information about the human's gaze. The posterior probabilities of the models are calculated through interacting model matched filtering. The candidate location with the highest posterior probability is chosen to be the estimate of the true goal location. Detailed quantitative evaluations of the G-MMIE algorithm on two different datasets involving 15 subjects, and comparisons with state-of-the-art intention inference algorithms are presented.

Keywords Human intention inference · Information fusion · Human-robot collaboration

H. Ravichandar
Department of Electrical and Computer Engineering, University of Connecticut, Storrs, CT 06269.
E-mail: harish.ravichandar@uconn.edu

A. Kumar
Department of Electrical and Computer Engineering, University of Connecticut, Storrs, CT 06269.
E-mail: avnish.kumar@uconn.edu

A. Dani
Department of Electrical and Computer Engineering, University of Connecticut, Storrs, CT 06269.
E-mail: ashwin.dani@uconn.edu

1 Introduction

Human intention inference is the first natural step for achieving safety in Human-Robot Collaboration (HRC) (Tsai et al 2014; Dani et al 2014). Studies in psychology suggest that when two humans interact, they infer each other's intended actions for safe interaction and collaboration (Baldwin and Baird 2001; Simon 1982). Taking inspiration from how humans interact with each other, the safety, operational efficiency, and task reliability in HRC could be greatly improved by providing robots with the capability to infer human intentions. For instance, in Warrier and Devasia (2017); Liu et al (2016); Li and Ge (2014), inference of the human partner's intention is shown to improve the overall performance of tasks requiring HRC. While the word "intention" is used to describe several characteristics in the context of HRC, we define intention as the goal location of reaching motions. In this work, we develop an algorithm to infer the intention of human partners. To this end, a Neural Network (NN) is used to learn the complex dynamics of human arm reaching motion and the learned model is used to infer the user's intentions. It is shown that humans and animals generate inher-

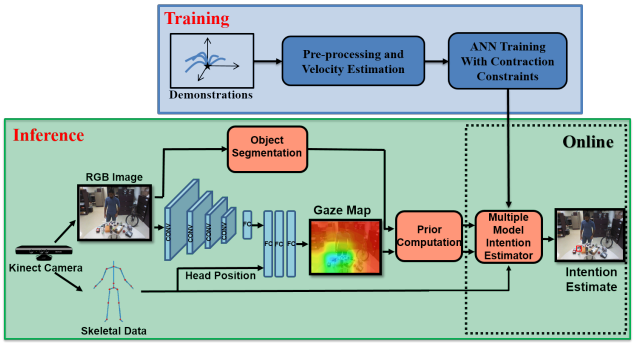

Figure 1 Block diagram illustrating the building blocks of the gaze-based multiple-model intention estimator (G-MMIE) algorithm.

ently closed-loop stable limb motions while performing reaching tasks (Schaal 1999). Hence, the problem of learning the arm dynamics is formulated as a parameter learning problem under constrains, derived using contraction analysis of nonlinear systems (Lohmiller and Slotine 1998), that aid in learning stable nonlinear dynamics. Details of the learning algorithm can be found in Ravichandar and Dani (2015b); Ravichandar et al (2017). Once the model is learned, intention inference could be carried out by using the multiple-model-intention estimator (MMIE) presented in Ravichandar and Dani (2015a). The MMIE algorithm uses an interacting multiple model (IMM) filtering approach in which the posterior probabilities of candidate goal locations are computed through model-matched filtering (cf. Bar-Shalom et al (2001); Granstrom et al (2015)).

However, the MMIE algorithm is not suitable for cluttered environments with a large number of candidate goal locations. This is due to the fact that the MMIE algorithm will have to consider a large number of models and run equally many filters in parallel to carry out the inference. A carefully designed prior distribution that is based on heuristics would render the MMIE algorithm suitable to such scenarios by reducing the number of candidate goal locations and, ultimately, the time taken to infer the true goal location. In this work, gaze information is used to compute the prior distribution.

Human gaze and attention are known to be task-dependent and goal-oriented (Yarbus 1967). Gaze cues are proved to be efficient in communicating attention (Kleinke 1986). In Flanagan and Johansson (2003), it is demonstrated that adults predict action goals by fixating on the end location of an action before it is reached, both when they execute an action themselves and when they observe someone else executing an action. In Gredebäck and Falck-Ytter (2015), a survey of various studies that followed Flanagan and Johansson (2003) is presented. In Hayhoe and Ballard (2005), it is noted that the point of fixation in a given scenario may not be the most visually salient location, but rather will correspond to the best location given the specifications and demands of the task. These studies in Yarbus (1967); Kleinke (1986); Flanagan and Johansson (2003); Gredebäck and Falck-Ytter (2015) suggest that the use of gaze information would be helpful in predicting the intention or goal location of human reaching motions.

The contribution of this work involves using gaze as a heuristic to compute the prior probabilities of the candidate goal locations in order to perform early prediction of the goal location of human reaching actions. The convolution neural network (CNN)-based gaze estimation algorithm, presented in Recasens et al (2015), is used to obtain a gaze map from a given RGB image. The gaze map is a spatial map that contains the probability of each pixel being the gaze point. Then, the top N_g candidate goal locations are chosen by thresholding and their prior probabilities are computed based on the gaze map. The prior distribution is subsequently used in a Bayesian setting to obtain a maximum-a-posteriori (MAP) estimate of the goal location.

Compared to an earlier version of the G-MMIE algorithm presented in Ravichandar et al (2016), this paper presents (1) a comprehensive literature review, (2) a detailed experimental evaluation, (3) quantitative comparisons with state-of-the-art intention inference algorithms, and (4) a thorough discussion of the results. The G-MMIE algorithm is evaluated by conducting four experiments on two different datasets containing data collected from 15 subjects.

Related Work

Algorithms for human-intention estimation are studied in various areas, such as human-computer interaction (Preece et al 1994) and human-robot interaction (HRI) (Goodrich and Schultz 2007). The inference of such intentions are carried out by gathering different modalities of information about the interaction, e.g., by using gestures (Matsumoto et al 1999), voice commands (Matuszek et al 2013), facial expressions (Bartlett et al 2003), characteristics of the objects in the workspace (Koppula et al 2013), human movement (Mainprice et al 2015), or by measuring physiological information, such as electromyography (Razin et al 2017), heart rate and skin response (Kulic and Croft 2007). Human intention inference has been studied by using multivariate Gaussians (Razin et al 2017), hidden Markov models (HMMs) (Ding et al 2011), dynamic Bayesian networks (DBNs) (Gehrig et al 2011), growing HMMs (GHMMs) (Elfring et al 2014), Conditional Random Fields (CRFs) (Koppula et al 2013), Gaussian

mixture models (GMMs) (Luo et al 2017), and neural networks (NNs) (Ravichandar and Dani 2017).

In Kulic and Croft (2007), an affective state estimation algorithm based on HMMs is presented. The affective state, represented using valence/arousal characteristics, is measured by using physiological signals, such as heart rate and skin response. The valence/arousal representation of human intention only indicates the degree of approval to a given stimulus. In Song et al (2013), human intention is predicted by visually observing the hand-object interaction in grasping tasks. This work is specific to grasping motions and predicts the required grasping configuration for a given task. In Strabala et al (2013), handover tasks are studied and the intention to handover an object is predicted by using key features extracted from the vision and the pose (position + orientation) data. The aforementioned works, however, do not consider any hueristics to compute a prior distribution to aid goal location prediction. In Pérez-D'Arpino and Shah (2015), task-level information is used to compute the prior distribution over the goal locations. Unlike Pérez-D'Arpino and Shah (2015), the G-MMIE algorithm leverages on gaze information to compute the prior distribution and is task-agnostic.

In Friesen and Rao (2011), a Bayesian gaze following method is introduced to explain how humans follow each other's gaze. Since gaze direction indicates whether the human is paying attention to the robot, many works have demonstrated the effectiveness of gaze for ensuring safety in HRI (Traver et al 2000; Matsumoto et al 1999). In Bader et al (2009), gaze is used to explain the causal relationships between natural gaze behavior and other input modalities or system states in manipulation tasks. In Hart et al (2014), nonverbal cues including gaze are studied for timing coordination between humans and robots. A detailed review of methods involving the use of gaze in HRI is presented in Admoni and Scassellati (2017). In contrast to the aforementioned studies that use gaze information, the G-MMIE algorithm infers the goal location of reaching motions by leveraging upon the availability of both human arm motion data and gaze information. Instead of using gaze to evaluate the attentiveness of the human or identifying the point of fixation, the G-MMIE algorithm computes the prior distribution of the goal location by using the gaze information as a heuristic.

2 Learning Contracting Nonlinear Dynamics of Human Reaching Motion

In this section, a method for learning the dynamics of human arm reaching motion is presented. Consider a state variable $x\left(t\right) \in \mathbb{R}^n$ and a set of $N_{\mathcal{D}}$ demonstrations $\{\mathcal{D}_i\}_{i=1}^{N_{\mathcal{D}}}$ representing reaching motions to various goal locations. Each demonstration would consist of the trajectories of the state $\{x\left(t\right)\}_{t=0}^{t=T}$ and the trajectories of the state derivative $\{\dot{x}\left(t\right)\}_{t=0}^{t=T}$ from time $t = 0$ to $t = T$. All state trajectories of the demonstrations are translated such that they converge to the origin. Let the translated demonstrations be solutions to the underlying dynamical system governed by the following first order differential equation

$$\dot{x}\left(t\right) = f\left(x\left(t\right)\right) + w\left(t\right) \tag{1}$$

where $f\colon \mathbb{R}^n \to \mathbb{R}^n$ is a nonlinear continuously differentiable function and $w \sim \mathcal{N}(0, Q_c)$ is a zero mean Gaussian process with covariance Q_c. Since all the trajectories of the translated demonstrations converge to the origin, the system defined in (1) could be seen as a globally contracting system. The nonlinear function $f\left(\cdot\right)$ is modeled by using a neural network (NN): $f\left(x(t)\right) = W^T\sigma\left(U^T s\left(t\right)\right) + \epsilon\left(s\left(t\right)\right)$ where $s\left(t\right) = \left[x\left(t\right)^T, 1\right]^T \in \mathbb{R}^{n+1}$ is the input vector to the NN, $U \in \mathbb{R}^{n+1\times n_h}$ and $W \in \mathbb{R}^{n_h\times n}$ are the bounded constant weight matrices, $\epsilon\left(s\left(t\right)\right) \in \mathbb{R}^n$ is the function reconstruction error that goes to zero after the NN is fully trained, n_h is the number of neurons in the hidden layer of the NN, $\sigma(U^T s\left(t\right)) = [\frac{1}{1+\exp(-(U^T s(t))_1)}, \cdots, \frac{1}{1+\exp(-(U^T s(t))_i)}, \cdots, \frac{1}{1+\exp(-(U^T s(t))_{n_h})}]^T$ is the vector-sigmoid activation function, and $\left(U^T s\left(t\right)\right)_i$ is the i^{th} element of the vector $\left(U^T s\left(t\right)\right)$. Note that only one NN is used to represent the dynamics of reaching motion trajectories that converge to the origin. Arm motion trajectories pertaining to different goal locations can be obtained by corresponding liner translations of the solutions to the dynamical system in (1).

The constrained optimization problem to be solved in order to train a contracting NN is given by

$$\{\hat{W}, \hat{U}\} = \arg\min_{W,U} \{\alpha E_D + \beta E_W\} \tag{2}$$

$$\text{such that } \frac{\partial f}{\partial x}^T M + M\frac{\partial f}{\partial x} \leq -\gamma M,\ M > 0 \tag{3}$$

where $E_D = \sum_{i=1}^{D} \left[y_i - a_i\right]^T \left[y_i - a_i\right]$, $y_i \in \mathbb{R}^n$ and $a_i \in \mathbb{R}^n$ represent the target and the network's output of the ith demonstration, E_W is the sum of the squares of the NN weights, $\alpha, \beta \in \mathbb{R}$ are scalar parameters of regularization (MacKay 1992), $\gamma \in \mathbb{R}$ is a strictly positive constant, and $M \in \mathbb{R}^{n\times n}$ represents a constant positive symmetric matrix. The Jacobian, $\frac{\partial f}{\partial x}$ is given by

$$\frac{\partial f}{\partial x} = W^T\frac{\partial\sigma\left(U^T s\right)}{\partial x} = W^T\left[\Sigma^{'}\left(U^T s\right)\right]U_x^T \tag{4}$$

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

$$\Sigma^{'}(b) = \mathrm{diag}\left(\sigma(b_1)(1-\sigma(b_1)),\right. \\ \left.\sigma(b_2)(1-\sigma(b_2)), \cdots, \sigma(b_p)(1-\sigma(b_p))\right), \quad (5)$$

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

3 Gaze-based Prior Computation

3.1 Gaze Map Estimation

This section breifly describes the CNN, introduced in Recasens et al (2015), that is used to extract gaze information from an RGB image. To this end, a deep architecture of CNN is employed. The input (features) to the CNN is a $D_w \times D_h$ RGB image of the subject looking at an object and the relative position of the subject's head in that image. The output is the gaze map $\mathcal{G}$ of size $D_w \times D_h$ containing the probabilities of each pixel being the gaze point.

Data: The dataset used for training the CNN model, as described in Recasens et al (2015), is created by concatenating images from six different sources: 1548 images from *SUN* (Xiao et al 2010); 33790 images from *MS coco* (Lin et al 2014); 9135 images from *Actions40* (Yao et al 2011); 7791 images from *PASCAL* (Everingham et al 2010); 508 images from the *ImageNet* detection challenge (Russakovsky et al 2015); and 198097 images from the *Places* dataset (Zhou et al 2014).

Implementation of Convolution Neural Network (CNN): The five layered CNN shown in Fig. 1 is implemented using Caffe (Jia et al 2014). Images of size $224 \times 224 \times 3$ are used for training the CNN. These input images are filtered by 96 convolution kernels of size $11 \times 11 \times 3$ and fed into the first convolution layer of size $55 \times 55 \times 96$. The output of the first layer is filtered with 256 convolution kernels of size $5 \times 5 \times 48$ and fed to the second convolution layer. The subsequent three layers are connected to one another without any pooling layers between them. The third convolution layer has 384 convolution kernels of size $3 \times 3 \times 256$ connected to the normalized and pooled outputs of the second convolution layer. The fourth convolution layer has 384 convolution kernels of size $3 \times 3 \times 192$ and the fifth convolution layer has 256 convolution kernels of size $3 \times 3 \times 192$. The remaining four layers used in the network are fully connected (FC) and are of sizes 100, 400, 200, and 169. See Recasens et al (2015) for a more in-depth description of the CNN framework.

3.2 Computation of Prior Distribution using Gaze Map

The average prior probability $\bar{p}_j(0)$ of the jth object in the scene is calculated as follows

$$\bar{p}_j(0) = \sum_{i \in \mathcal{G}P_j} \left(\mathcal{G}(i)/NP_j\right), \quad (6)$$

where $\mathcal{G}(i)$ is the probability of the ith pixel being the gaze point, NP_j is the number of pixels associated with the jth object, and $\mathcal{G}P_j$ is the set of all pixel locations associated with the jth object. Of all the objects in the scene, the objects that correspond to the top N_g average prior probabilities are chosen as the candidate goal locations. The prior probability of each of the N_g candidate locations being the goal location is calculated as follows

$$\mu_j(0) = \frac{\sum_{i \in \mathcal{G}P_j} \left(\mathcal{G}(i)/NP_j\right)}{\sum_{j=1}^{N_g} \sum_{i \in \mathcal{G}P_j} \left(\mathcal{G}(i)/NP_j\right)} \quad (7)$$

where $\mu_j(0)$ is the prior probability of g_j being the goal location and g_j refers to the location of the jth object.

4 Intention Inference using Gaze Prior and Motion Dynamics

Given the trained network and a trajectory of the reaching hand, the problem involves inferring the goal location ahead in time. Let a finite set of candidate goal locations that the human can reach be $G = \{g_1, g_2, \cdots \cdot\cdot, g_{N_g}\}$. Let the equilibrium point of the NN be the origin ($x = 0_{n\times 1}$). The NN learned from human demonstrations is used to represent human motion. For each goal location g_j, the state vector and the corresponding dynamics are defined as $x^j(t) = [[x_{pos}(t) - g_j]^T, x_{vel}^T(t)]^T$ and $\dot{x}^j(t) = f(x^j(t))$. Similarly, for a set of N_g goal locations, a set of N_g dynamic systems is formed. The discretized versions of these systems are given by

$$x^j(k+1) = x^j(k) + T_s f\left(x^j(k)\right) + T_s w(k) \quad (8)$$

where $j = 1, .., N_g$ and T_s is the sampling period. The measurement model is given by

$$z(k) = h(x(k)) + v(k), \qquad j = 1, 2, ..., N_g \quad (9)$$

where $z(k)$ is the measurement vector at time instant k, $v \sim \mathcal{N}(0, R)$ is a zero mean Gaussian process with covariance R and $h(x(k)) = \begin{bmatrix} 1&0&0&0&0&0 \\ 0&0&1&0&0&0 \\ 0&0&0&0&1&0 \end{bmatrix} x(k)$ is the measurement function.

Let $M_1, M_2, ..., M_{N_g}$ represent the N_g models defined in (8) and (9) for the set of candidate goal locations G, the objective is to estimate $p(g_j|Z_{1:k})$. The

expression $p(g_j|Z_{1:k})$ denotes the posterior probability of each g_j being the actual goal location given a set of k measurements $Z_{1:k} = [z(1), z(2), \cdots, z(k)]$. Note that $p(g_j|Z_{1:k}) = p(M_j|Z_{1:k})$ since the models and goal locations have a one-to-one correspondence. Hence, in order to obtain the posterior probabilities $p(g_j|Z_{1:k})$, $j = 1, .., N_g$, the posterior probabilities of the models $p(M_j|Z_{1:k})$, $j = 1, .., N_g$ is computed. The posterior probability $p(M_j|Z_{1:k})$ is calculated using the Bayes' theorem as follows $p(M_j|Z_{1:k}) = \frac{p(Z_{1:k}|M_j)p(M_j)}{\sum_{i=1}^{N} p(Z_{1:k}|M_i)p(M_i)}$, where $p(Z_{1:k}|M_j)$ is the likelihood of M_j, and $p(M_j)$ is the prior probability of M_j. In the IMM framework with N_g models, the likelihood functions $p(Z_{1:k}|M_j)$ are computed using N_g filters running in parallel. The G-MMIE algorithm uses extended Kalman filters (EKFs). Each iteration of the IMM filter for intention inference is divided into three main steps (cf. Bar-Shalom et al (2001)). These steps are summarized in the remainder of this subsection.

Interaction/Mixing: At the beginning of each iteration, the initial conditions (state estimate $\hat{x}^{0j}(k-1|k-1)$ and covariance $\hat{P}^{0j}(k-1|k-1)$), 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

$$\hat{x}^{0j}(k-1|k-1) = \sum_{i=1}^{N_g} \hat{x}^{i}(k-1|k-1) \times \mu_{i|j}(k-1|k-1),\ j = 1, .., N_g \quad (10)$$

$$\hat{P}^{0j}(k-1|k-1) = \sum_{i=1}^{N_g} \mu_{i|j}(k-1|k-1) \{\hat{P}^{i}(k-1|k-1) + [\hat{x}^{i}(k-1|k-1) - \hat{x}^{0j}(k-1|k-1)] \times [\hat{x}^{i}(k-1|k-1) - \hat{x}^{0j}(k-1|k-1)]^T\},\ j = 1, .., N_g \quad (11)$$

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

$$\mu_{i|j}(k-1|k-1) = \frac{\Pi_{ij}\mu_i(k-1)}{\bar{c}_j},\ i, j = 1, 2, ..., N_g \quad (12)$$

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

Model Matched Filtering: Once the initial conditions $\hat{x}^{0j}(k-1|k-1)$ and $\hat{P}^{0j}(k-1|k-1)$ 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 $\Lambda_j(k)$ are computed using the mixed initial condition (10) and the corresponding covariance (11). The likelihood $\Lambda_j(k)$, a Gaussian distribution with the predicted measurement as the mean and the covariance equal to the innovation covariance, is given by

$$\begin{aligned}\Lambda_j(k) =& p(z(k)|M_j(k), \hat{x}^{0j}(k-1|k-1), \hat{P}^{0j}(k-1|k-1)) \\ =& \mathcal{N}(z(k); \hat{z}^{j}(k|k-1; \hat{x}^{0j}(k-1|k-1)), S^{j}(k; \hat{P}^{0j}(k-1|k-1))),\ j = 1, .., N_g \end{aligned} \quad (13)$$

where $S^{j}(k; P^{0j}(k-1|k-1))$ is the innovation covariance and $\hat{z}^{j}(k|k-1; \hat{x}^{0j}(k-1|k-1))$ is the jth filter's predicted measurement at time k.

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

$$\begin{aligned}\mu_j(k) &= p(g_i|Z_{1:k}) = p(M_j(k)|Z_{1:k}) \\ \mu_j(k) &= p(z(k)|M_j(k), Z_{1:k-1})p(M_j(k)|Z_{1:k-1}) \\ \mu_j(k) &= \frac{\Lambda_j(k)\bar{c}_j}{\sum_{j=1}^{N} \Lambda_j(k)\bar{c}_j}, \qquad j = 1, 2, ..., N_g \end{aligned} \quad (14)$$

and the goal location estimate $\hat{g}(k)$ is given by

$$\hat{g}(k) = \arg\max_{g \in G}\ p(g|Z_{1:k}) \quad (15)$$

The optimization problem in (15) is solved by choosing the location $g_i \in G$ corresponding to the model M_i with the highest model probability $\mu_i(k)$ at time k.

Model Switch Detection: Tasks with a sequence of reaching motions involve switching between consecutive reaching motions. In such tasks, the switching time instants are not known *a priori*. The G-MMIE algorithm is able to detect the switches on-the-fly by observing the model probabilities. After a reaching motion is complete, a change in the goal location estimate indicates that the next reaching motion has begun. Once the switch is detected, the gaze prior computed from the current image is used to select the top N_g objects. The model probabilities are reinitialized to the newly obtained gaze-prior. The implementation details of the inference algorithm is given in Algorithm 1.

5 Experimental Validation

In order to validate the G-MMIE algorithm, a total of four experiments are carried out. The algorithm is coded in MATLAB and is run on a standard desktop computer running Intel i7 processor with 8 Gigabytes

Figure 2 Gaze-based prior computation: The input image (left), the gaze map overlaid on the input image (center), and the computed prior probabilities of the top five objects (right).

of memory. The average computation time of the G-MMIE algorithm for processing each frame and giving out an estimate is 0.053 sec. The average computation time is computed over a 100 sample trajectories. In all experiments, the training and testing data are mutually exclusive. In the training set, each trajectory is labeled based on the ground truth goal location. Note that the ground truth labeling is done only for the trajectories in the training set. The computation of gaze-based priors for a sample trajectory from Experiment 2 is illustrated in Fig. 2. The objectives of each experiment is described below:

- *Experiment 1*: Evaluate and compare the ability of the G-MMIE algorithm to infer intentions with that of state-of-the-art inference algorithms.
- *Experiment 2*: Evaluate the importance of gaze prior in the G-MMIE algorithm in a cluttered scenario with a large number of objects.
- *Experiment 3:* Test the ability of the G-MMIE algorithm to infer the goal locations of sequences of reaching motions.
- *Experiment 4:* Evaluate the ability of the G-MMIE algorithm to infer sub-activity labels on an independent dataset by using the Cornell's CAD-120 dataset (Koppula et al 2013).

In all the experiments, the position and velocity of the hand in 3-dimensional (3D) Cartesian space are considered to be the elements of the state vector $x(k) \in \mathbb{R}^6$. The initial state estimate covariance $\hat{P}^j(0)$, $j = 1, 2, \cdot\cdot, N_g$, the process noise covariance Q^j , $j = 1, \cdot\cdot, N_g$, and the measurement noise covariance R are selected to be $0.2I_{6\times6}$, $0.1I_{6\times6}$, and $0.2I_{6\times6}$, respectively. The state estimates $\hat{x}^j$, $j = 1, 2, \cdot\cdot, N_g$, are initialized using the first two measurement $z(1)$ and $z(2)$ (a finite difference method is used for the velocity initialization). The diagonal and off-diagonal elements of the model transition matrix are chosen to be $\Pi_{ii}(m, m) = 1 - 0.01(N_g - 1)$ and $\Pi_{ij}(m, n) = 0.01, \forall m \neq n$, respectively, where $\Pi_{ij}(m, n)$ is the mnth element of Π_{ij}. The number of goal locations to be considered (after thresholding based on gaze priors) is empirically chosen be $N_g = \min(\tilde{N}_g, 5)$ where $\tilde{N}_g$ is the total number of objects in the scene.

5.1 Experiment 1

In this experiment, the G-MMIE algorithm is evaluated on a testing data set comprised of a total of 1050 trajectories of reaching motions collected from 11 different subjects at 30 Hz using a Microsoft Kinect in the Robotics and Controls Lab at UConn. A separate set of 10 reaching trajectories from one of the 11 subjects is used to train the motion model. The number of neurons in the hidden layer is chosen to be 50. The subjects are given no other instructions but to reach for the objects. The test trajectories are collected in scenarios with a variety of initial positions, distinct motion profiles, different object locations, and the number of subjects simultaneously performing reaching motions. Further, the number of objects on the table varied between 4 and 10. The total number of frames for each trajectory is not fixed and the intended object is reached at varying frame numbers (each trajectory contained roughly 100 to 150 frames of tracked skeletal data). The hand position data obtained from the subjects are processed to obtain the velocity and acceleration estimates using a Kalman filter (see Morato et al (2014) for details). An exemplary sequence of images, illustrating the goal locations inferred by the G-MMIE algorithm as two subjects simultaneously perform reaching motions, is shown in Fig. 3. Note that when two subjects reach for their respective goal locations simultaneously, two separate instances of the G-MMIE algorithm (one for each subject) are used to infer intentions. Further, each reaching motion of each subject is counted as a separate run.

Furthermore, the performance of the G-MMIE algorithm is compared with that of the MMIE (Ravichandar and Dani 2015a), the adaptive neural intention estima-

Figure 3 Sequence of images (left to right) showing the intentions simultaneously inferred by the G-MMIE algorithm as two subjects reach for their respective objects. The prior probabilities of the top 5 objects for both subjects (computed based on their respective gaze maps) are overlaid on the first frame.

Algorithm 1: The G-MMIE algorithm

Observe the work space to estimate the set of all goal locations;
Compute the gaze map $\mathcal{G}$ of the first image using the trained CNN;
Compute the average prior probability of all the objects in the scene using (6);
Choose the top N_g objects based on their average prior probabilities as candidate goal locations;
Compute the prior probability distribution among the N_g objects using (7) Obtain N_g models by translating the original contracting system to all candidate goal locations using (8);
Initialize $\hat{x}^j(0), \hat{P}^j(0),\ j = 1, 2, \cdots, N_g$ and $\hat{g}(0)$;
Define the parameters of the system: $R, \Pi_{ij}, Q^j, i, j = 1, 2, \cdot\cdot, N_g$;

while *data for the current time step is present* **do**
  Read the current measurement $z(k)$

  Mixing Probabilities:
  Using previous state estimates and covariances $\hat{x}^j(k-1), \hat{P}^j(k-1),\ j = 1, 2, \cdots, N_g$, compute the mixed initial estimates and covariances $\hat{x}^{0j}(k-1|k-1), \hat{P}^{0j}(k-1|k-1),\ j = 1, 2, \cdots, N_g$ based on (10) and (11);

  Model Matched Filtering:
  From the mixed initial estimates and the corresponding covariances from the previous step, compute the state model likelihoods $\Lambda_j(t)$, state estimates and covariances $\hat{x}^j(k), \hat{P}^j(k),\ j = 1, 2, \cdots, N_g$ using extended Kalman filters (EKFs) and (13);

  Model Probabilities:
  Compute the posterior probability of each model $p(M_j|Z_{1:k})$ using (14) based on the output of the previous step ;
  Infer the goal location (intention) $\hat{g}(k)$ using (15);

  Model Switch Detection:
  if *Object is reached* **then**
    if *model switch is detected* **then**
      Compute the gaze map $\mathcal{G}$ of the current image and start over;
    end
  end
end

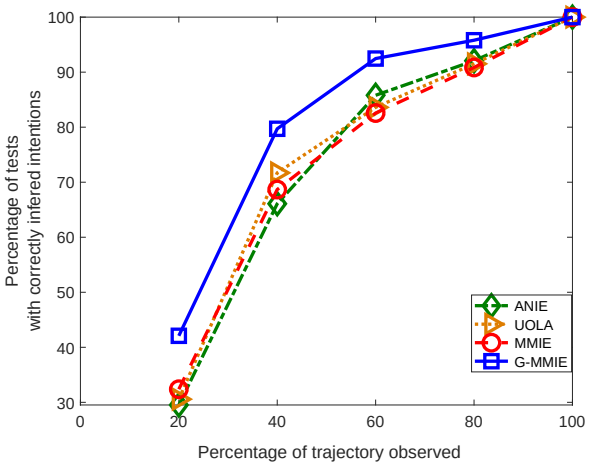

Figure 4 Percentage of tests with correctly inferred intention as a function of the percentage of trajectory observed for Experiment 1.

tor (ANIE) (Ravichandar and Dani 2017), and the unsupervised online learning algorithm (UOLA) (Luo et al 2017). In Fig. 4, the percentages of tests where the intention is correctly inferred by different algorithms are shown as a function of the percentage of trajectory observed.

5.2 Experiment 2

In Experiment 2, the importance of gaze prior is illustrated by considering a scenario with a large number of objects. For this experiment, a total of 24 objects are placed close to each other in a cluttered manner. No additional offline training is performed and the same models trained in Experiment 1 (using 10 trajectories collected from one subject) is used in the evaluations. A new set of 94 reaching trajectories are collected from two subjects as part of the testing set. The data are collected from the two subjects on separate occasions. Similar to Experiment 1, the data are collected at 30 Hz using a Microsoft Kinect in the Robotics and Controls Lab at UConn. The initial conditions, motion profiles, and object locations of the trajectories in the data set are different. The recorded joint position data are fil-

tered to obtain the velocity and acceleration estimates using a Kalman filter.

To illustrate the advantage of the gaze prior, a sequence of images with the goal locations inferred by the MMIE algorithm (Ravichandar and Dani 2015a) and the G-MMIE algorithm is shown in Fig. 5. In Fig. 6, the percentages of tests where the intention is correctly inferred by different algorithms are shown as a function of the percentage of trajectory observed.

5.3 Experiment 3

In this experiment, the G-MMIE algorithm is tested on sequences of reaching motions in two testing scenarios. In the first testing scenario, a set of five sequences, each with four reaching motions, is collected from one subject. In this testing scenario, a total of 15 objects are randomly placed close to each other in a cluttered manner. The second testing scenario involved two subjects collaborating to assemble a desk drawer. The assembly involved 15 reaching motions and 6 candidate goal locations. No additional offline training is performed and the same models trained in Experiment 1 (using 10 trajectories collected from one subject) is used in the evaluations.

The G-MMIE algorithm is found to detect the model switch in all occasions of both the testing scenarios. In Fig. 7, a sequence of images with the goal locations inferred by the G-MMIE algorithm in the first testing scenario is shown. A similar sequence of images for the second testing scenario is shown in Fig 8.

5.4 Experiment 4

Modifications to the CAD-120 dataset: For the purpose of using goal location prediction to identify sub-activity labels, modifications to the CAD-120 dataset are made following the steps described in Monfort et al (2015). There are 10 sub-activities in the original CAD-120 dataset: *reaching*, *moving*, *pouring*, *eating*, *drinking*, *opening*, *placing*, *closing*, *cleaning*, and *null*. Each sub-activity is considered to be associated with a goal location (i.e., a one-to-one mapping between goal locations and sub-activity labels). The *moving* sub-activity is considered to be a part of the succeeding sub-activity and is merged with the following sub-activity. The *null* sub-activity is ignored since it is not driven by a goal location. The *opening* sub-activity is divided into two new sub-activities, namely, *opening-the-microwave* and *opening-a-jar* since they have different goal locations. These modifications result in a total of nine sub-activities. The goal locations of each sub-activities is computed by averaging over the observed goal locations in the training set.

Table 1 Results of labeling sub-activities in the CAD-120 dataset

Algorithm	Accuracy	Macro Precision	Macro Recall
G-MMIE 20% sequence	67.25	55.32 ± 16.76	52.7 ± 16.17
G-MMIE 40% sequence	83.98	76.16 ± 12.62	73.33 ± 15.81
G-MMIE 60% sequence	95.54	98.95 ± 5.23	92.63 ± 6.43
G-MMIE 80% sequence	97.66	99.54 ± 0.59	96.9 ± 1.44
G-MMIE 100% sequence	100	100 ± 0.0	100 ± 0.0
ANIE 20% sequence	58.21	40.34 ± 21.95	37.04 ± 27.67
ANIE 40% sequence	76.12	70.02 ± 26.28	63.22 ± 28.89
ANIE 60% sequence	92.54	97.57 ± 4.81	86.3 ± 13.56
ANIE 80% sequence	95.52	98.59 ± 0.28	92.22 ± 11.76
ANIE 100% sequence	100	100 ± 0.0	100 ± 0.0
I-LQR 20% sequence	80.9	65.0 ± 3.1	77.3 ± 2.4
I-LQR 40% sequence	82.5	73.4 ± 2.2	91.4 ± 0.6
I-LQR 60% sequence	84.1	79.1 ± 2.5	94.2 ± 0.6
I-LQR 80% sequence	90.4	87.5 ± 1.8	96.2 ± 0.3
I-LQR 100% sequence	100	100 ± 0.0	100 ± 0.0
ATCRF 100% sequence	86.0	84.2 ± 1.3	76.9 ± 2.6

Performance Evaluation: The dataset is randomly divided into testing and training set with 10% of trajectories allocated for the test set. The G-MMIE algorithm's effectiveness in inferring the sub-activity label (measured in terms of accuracy, precision, and recall) is evaluated at different percentages of trajectory that is observed (20%, 40%, 60%, 80%, and 100%). An exemplary sequence of images, illustrating the goal locations inferred by the G-MMIE algorithm as a subject performs two reaching motions (one with each hand), is shown in Fig. 9. In this experiment, the G-MMIE algorithm is compared with the MMIE (Ravichandar and Dani 2015a), ANIE (Ravichandar and Dani 2017), I-LQR (Monfort et al 2015), and ATCRF (Koppula et al 2013) algorithms. The results of the comparison are summarized in Table 1. The statistics in Table 1, pertaining to the ANIE, I-LQR and the ATCRF algorithms, reported in Ravichandar and Dani (2017), are used here for comparison. Accuracy is given by $\frac{N_C}{N_T}$ where N_C denotes the number of correct classifications and N_T is the total number of classifications. Precision and recall are given by $\frac{N_{TP}}{N_{TP}+N_{FP}}$ and $\frac{N_{TP}}{N_{TP}+N_{FN}}$, respectively, where N_{TP} denotes the number of true positives, N_{FP} denotes the number of false positives, and N_{FN} denotes the number of false negatives.

6 Discussion

This section provides a discussion of the experimental results presented in Section 5.

Experiment 1: In Experiment 1, the motion model used in the G-MMIE algorithm is trained from 10 trajectories collected from a single subject. The testing set consists of 1050 trajectories, that are different from the training trajectories, collected from 11 subjects. The trajectories in the test set varied in terms of several characteristics, such as motion profiles, number of

Figure 5 Image sequence showing the intention inferred by the MMIE algorithm (dashed green box) and the G-MMIE algorithm (solid red box) as a subject reaches for an object. A total of 24 objects are arbitrarily placed close to each other in a cluttered manner. The MMIE algorithm does not have an estimate for the initial frame as all the objects are assigned equal prior probability.

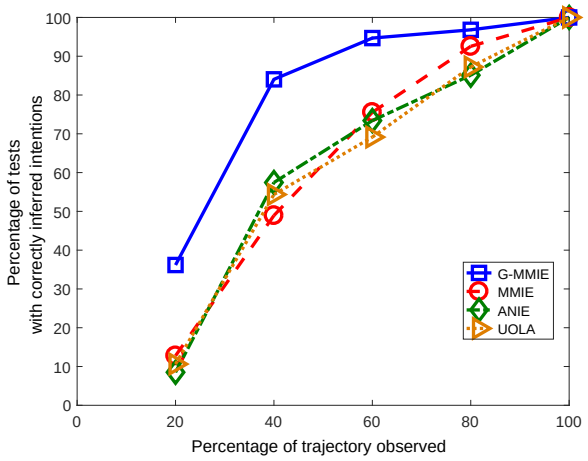

Figure 6 Percentage of tests with correctly inferred intention as a function of the percentage of trajectory observed for Experiment 2.

subjects simultaneously performing reaching motions, number of objects (goal locations), and placement of objects. The G-MMIE algorithm is shown to successfully predict the goal location of 79.71% of the trajectories in the test set after observing 40% of the subjects' arm motions. This accuracy increases to 92.47% after observing 60% of the subjects' arm motions. The comparison of the G-MMIE algorithm's performance with that of the MMIE, ANIE, and UOLA algorithms is shown in Fig. 4. After observing 20% of the trajectory, the G-MMIE algorithm performs 9.71% better than MMIE, 12.57% better than ANIE, and 11.52% better than UOLA. After observing 40%, the G-MMIE algorithm performs 11.04% better than MMIE, 13.62% better than ANIE, and 8% better than UOLA. After observing 60%, the G-MMIE algorithm performs 9.9% better than MMIE, 6.67% better than ANIE, and 8.86% better than UOLA. Finally, after observing 80%, the G-MMIE algorithm performs 4.95% better than MMIE, 3.73% better than ANIE, and 4.28% better than UOLA. This observation is to be expected since the G-MMIE takes advantage of the the gaze information to compute a prior distribution over the candidate goal locations.

Experiment 2: Examination of the results of Experiment 2 reveals the importance of the gaze prior in the G-MMIE algorithm. The experiment involved two subjects performing reaching motions in front of a table with a large number of objects placed in a cluttered manner. No part of the data collected for this experiment is used to train the motion model used by the G-MMIE. The comparison of the G-MMIE algorithm's performance with that of the MMIE, ANIE, and UOLA algorithms is shown in Fig. 6. The G-MMIE algorithm is shown to predict the correct goal locations earlier than the MMIE, ANIE, and UOLA algorithms. Specifically, after observing 20% of the trajectory, the G-MMIE algorithm performs 23.41% better than MMIE, 27.66% better than ANIE, and 25.54% better than UOLA. After observing 40%, the G-MMIE algorithm performs 35.11% better than MMIE, 26.6% better than ANIE, and 29.69% better than UOLA. After observing 60%, the G-MMIE algorithm performs 19.15% better than MMIE, 21.28% better than ANIE, and 25.53% better than UOLA. Finally, after observing 80%, the G-MMIE algorithm performs 4.26% better than MMIE, 11.71% better than ANIE, and 9.58% better than UOLA. The improvement in the intention inference is justified since the subjects are more likely to look directly at the object they want to reach for the test scenario consisting of 24 objects placed in a cluttered manner. Thus, using gaze to compute a prior distribution provides valuable information about the true goal location. Further, the gaze priors aid G-MMIE in reducing the number of candidate goal locations, thereby increasing the odds of accurate inference.

Experiment 3: The third experiment serves to validate the G-MMIE algorithm's ability to infer the goal locations in a sequence of reaching motions. As a subject starts moving towards the first goal, the probability associated with the first goal increases. Later, after reaching the goal location, as the subject starts to move towards the next goal, probability associated with the first goal decreases and that with the next goal location increases. This change is considered as a "goal

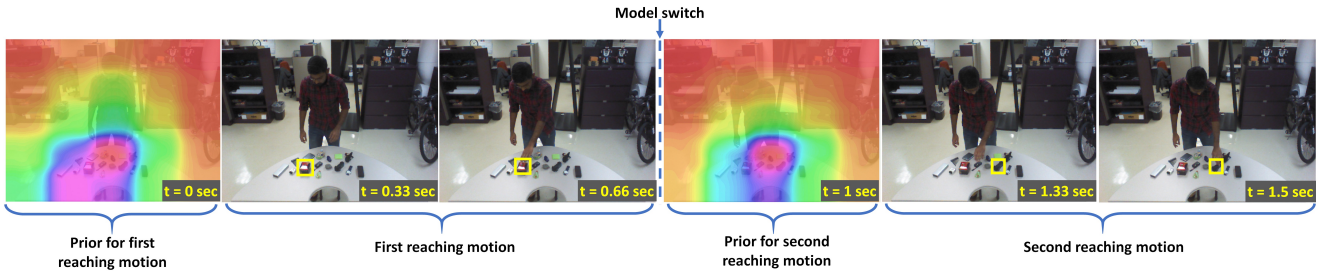

Figure 7 Image sequence showing the intention inferred by the G-MMIE algorithm (solid yellow box) as a subject reaches for two different objects in a sequence in the first testing scenario of Experiment 3. The fourth image shows the instant where the model switch is detected and the gaze map is computed to reinitialize the model probabilities.

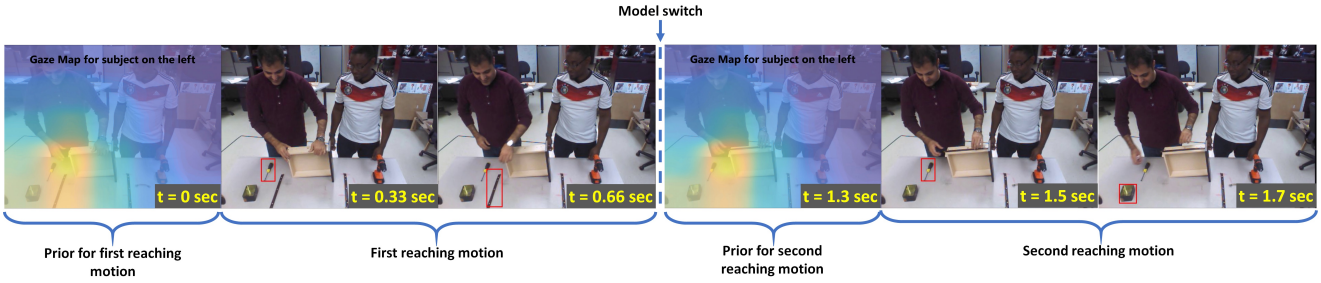

Figure 8 Image sequence showing the intention inferred by the G-MMIE algorithm (solid yellow box) as the subject on the left reaches for two different objects in sequence during a desk drawer assembly. The fourth image shows the instant where the model switch is detected and the gaze map is computed to reinitialize the model probabilities.

Figure 9 Sequence of images (left to right) showing the intentions inferred by the G-MMIE algorithm as Subject #4 from the CAD-120 dataset starts to reach for one object with his right hand and then reaches for another object with his left. The prior probabilities of the top 5 objects computed at the begining of both reaching motions are overlaid on the corresponding frames.

switch" and the probabilities of the goal locations are reinitialized by computing the gaze-based priors. The two testing scenarios used in this experiment simulate instances where one or more subject(s) perform(s) a series of reaching motions in order to collaboratively accomplish a task. It is crucial in such scenarios that the algorithm is capable of sequentially making correct predictions and recognizing the switches from one motion to the next. The G-MMIE algorithm successfully identifies the model switch and predicts the correct goal locations on all occasions.

It is also observed that on a few occasions, the subject seems to not look directly at the goal location at the beginning of the motion. In these scenarios, while the gaze prior might not be helpful, the G-MMIE algorithm is able to make the correct prediction as the subject's arm movement is taken into account. The gaze prior is effective because people, more often than not, tend to look at the object for which they are reaching as evidenced by other studies in literature, such as Flanagan and Johansson (2003); Gredebäck and Falck-Ytter (2015). Our experimental results also provide a strong evidence towards the effectiveness of the gaze prior even in the presence of scenarios where a person may not be exactly looking at the object they are reaching for. A sequence of images illustrating how the G-MMIE algorithm overcomes this challenge is shown in Fig. 10.

Experiment 4: In the final experiment, the G-MMIE algorithm's ability to classify sub-activities on the CAD-120 dataset is evaluated. The goal location predictions of the G-MMIE algorithm are translated to corresponding sub-activity labels. The comparison results indicate that the G-MMIE and I-LQR algorithms perform more accurately than the ANIE and ATCRF

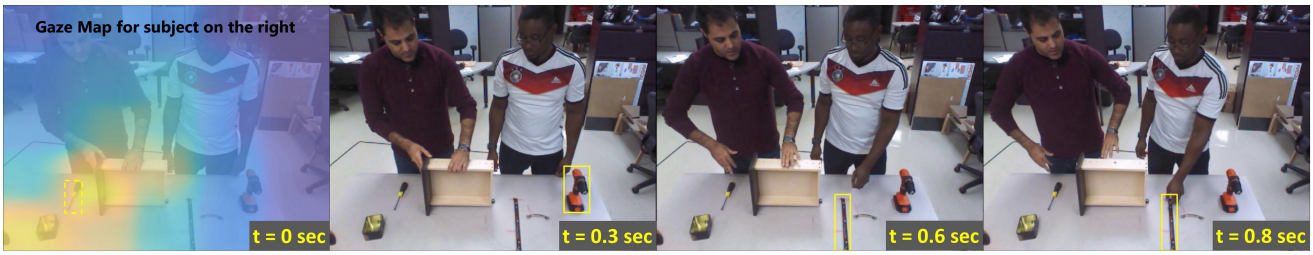

Figure 10 Sequence of images (left to right) showing how the G-MMIE algorithm performs during one of the few occasions when a subject is not looking in the direction of the goal location. While the gaze map does not provide much information about the true goal location (the candidate with the highest gaze prior is indicated in the first image with a dashed yellow box), the arm motion data aids the G-MMIE algorithm to make the correct prediction (indicated by solid yellow boxes).

algorithms. Specifically, the G-MMIE and I-LQR algorithms achieve 67.5% and 80.9% accuracy after 20% of the trajectory is observed, and 83.98% and 82.5% accuracy after 40% is observed, respectively. In contrast, the ANIE algorithm achieves 58.21% accuracy after 20% of the trajectory is observed, and 76.12% accuracy after 40% is observed. The ATCRF algorithm, on the other hand, achieves 86% accuracy after the entire trajectory is observed. This is likely due to the fact that both G-MMIE and I-LQR algorithms are maximum a posteriori (MAP) estimators with heuristically designed priors, and thus compute better initial guesses compared to ANIE and ATCRF algorithms.

7 Conclusion

The gaze-based multiple model intention estimator (G-MMIE), to infer the goal locations of human reaching motions, is presented. The goal location is estimated using a multiple-model Bayesian framework. The prior probability distribution of the goal location is computed based on information about the subject's gaze. The candidate goal location with the highest posterior probability is chosen to be the estimate of the algorithm. A set of four experiments conducted on two different datasets (one collected in-house and one independent) with data collected from 15 subjects is used to validate the G-MMIE algorithm. Owing to the advantage of using gaze-based prior distribution, the G-MMIE algorithm, on average, performs better than state-of-the-art intention inference algorithms in terms of early prediction of goal locations in a variety of scenarios. The G-MMIE algorithm is also shown to be capable of accurately inferring sub-activities in the CAD-120 dataset.

References

Admoni H, Scassellati B (2017) Social eye gaze in human-robot interaction: A review. Journal of Human-Robot Interaction 6(1):25–63

Bader T, Vogelgesang M, Klaus E (2009) Multimodal integration of natural gaze behavior for intention recognition during object manipulation. In: International conference on Multimodal interfaces, pp 199–206

Baldwin DA, Baird JA (2001) Discerning intentions in dynamic human action. Trends in cognitive sciences 5(4):171–178

Bar-Shalom Y, Li XR, Kirubarajan T (2001) Estimation with Applications to Tracking and Navigation. John Wiley and Sons

Bartlett MS, Littlewort G, Fasel I, Movellan JR (2003) Real time face detection and facial expression recognition: Development and applications to human computer interaction. In: IEEE Conference on Computer Vision and Pattern Recognition Workshop (CVPRW)., vol 5, pp 53–53

Dani AP, McCourt M, Curtis JW, Mehta S (2014) Information fusion in human-robot collaboration using neural network representation. In: IEEE Conference on Systems, Man, Cybernetics, pp 2114–2120

Ding H, Reißig G, Wijaya K, Bortot D, Bengler K, Stursberg O (2011) Human arm motion modeling and long-term prediction for safe and efficient human-robot-interaction. In: IEEE International Conference on Robotics and Automation, pp 5875–5880

Elfring J, Van De Molengraft R, Steinbuch M (2014) Learning intentions for improved human motion prediction. Robotics and Autonomous Systems 62(4):591–602

Everingham M, Van Gool L, Williams CK, Winn J, Zisserman A (2010) The pascal visual object classes (voc) challenge. International journal of computer vision 88(2):303–338

Flanagan JR, Johansson RS (2003) Action plans used in action observation. Nature 424(6950):769–771

Friesen AL, Rao RP (2011) Gaze following as goal inference: A bayesian model. In: Annual Conference of the Cognitive Science Society, vol 33

Gehrig D, Krauthausen P, Rybok L, Kuehne H, Hanebeck UD, Schultz T, Stiefelhagen R (2011) Combined intention, activity, and motion recognition for a humanoid household robot. In: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp 4819–4825

Goodrich MA, Schultz AC (2007) Human-robot interaction: a survey. Foundations and trends in human-computer interaction 1(3):203–275

Granstrom K, Willett P, Bar-Shalom Y (2015) Systematic approach to IMM mixing for unequal dimension states. IEEE Transactions on Aerospace and Electronic Systems 51(4):2975–2986

Gredebäck G, Falck-Ytter T (2015) Eye movements during action observation. Perspectives on Psychological Science 10(5):591–598

Hart JW, Gleeson B, Pan M, Moon A, MacLean K, Croft E (2014) Gesture, gaze, touch, and hesitation: Timing cues for collaborative work. In: HRI Workshop on Timing in Human-Robot Interaction, Bielefeld, Germany

Hayhoe M, Ballard D (2005) Eye movements in natural behavior. Trends in cognitive sciences 9(4):188–194

Jia Y, Shelhamer E, Donahue J, Karayev S, Long J, Girshick R, Guadarrama S, Darrell T (2014) Caffe: Convolutional architecture for fast feature embedding. In: ACM International Conference on Multimedia, pp 675–678

Kleinke CL (1986) Gaze and eye contact: a research review. Psychological bulletin 100(1):78

Koppula HS, Gupta R, Saxena A (2013) Learning human activities and object affordances from rgb-d videos. The International Journal of Robotics Research 32(8):951–970

Kulic D, Croft EA (2007) Affective state estimation for human–robot interaction. IEEE Transactions on Robotics 23(5):991–1000

Li Y, Ge S (2014) Human-robot collaboration based on motion intention estimation. IEEE/ASME Transactions on Mechatronics 19(3):1007–1014

Lin T, Maire M, Belongie S, Hays J, Perona P, Ramanan D, Dollár P, Zitnick CL (2014) Microsoft coco: Common objects in context. In: European Conference on Computer Vision, pp 740–755

Liu C, Hamrick JB, Fisac JF, Dragan AD, Hedrick JK, Sastry SS, Griffiths TL (2016) Goal inference improves objective and perceived performance in human-robot collaboration. In: International Conference on Autonomous Agents & Multiagent Systems, International Foundation for Autonomous Agents and Multiagent Systems, pp 940–948

Lohmiller W, Slotine JJE (1998) On contraction analysis for nonlinear systems. Automatica 34(6):683–696

Luo R, Hayne R, Berenson D (2017) Unsupervised early prediction of human reaching for human–robot collaboration in shared workspaces. Autonomous Robots pp 1–18

MacKay DJ (1992) Bayesian interpolation. Neural computation 4(3):415–447

Mainprice J, Hayne R, Berenson D (2015) Predicting human reaching motion in collaborative tasks using inverse optimal control and iterative re-planning. In: IEEE International Conference on Robotics and Automation (ICRA), pp 885–892

Matsumoto Y, Heinzmann J, Zelinsky A (1999) The essential components of human-friendly robot systems. In: International Conference on Field and Service Robotics, pp 43–51

Matuszek C, Herbst E, Zettlemoyer L, Fox D (2013) Learning to parse natural language commands to a robot control system. In: Experimental Robotics, Springer, pp 403–415

Monfort M, Liu A, Ziebart BD (2015) Intent prediction and trajectory forecasting via predictive inverse linear-quadratic regulation. In: AAAI Conference on Artificial Intelligence, pp 3672–3678

Morato C, Kaipa KN, Zhao B, Gupta SK (2014) Toward safe human robot collaboration by using multiple kinects based real-time human tracking. Journal of Computing and Information Science in Engineering 14(1):011,006–1–011,006–9

Pérez-D'Arpino C, Shah JA (2015) Fast target prediction of human reaching motion for cooperative human-robot manipulation tasks using time series classification. In: IEEE International Conference on Robotics and Automation (ICRA), pp 6175–6182

Preece J, Rogers Y, Sharp H, Benyon D, Holland S, Carey T (1994) Human-computer interaction. Addison-Wesley Longman Ltd.

Ravichandar H, Dani AP (2015a) Human intention inference through interacting multiple model filtering. In: IEEE Conference on Multisensor Fusion and Integration (MFI), pp 220–225

Ravichandar H, Dani AP (2015b) Learning contracting nonlinear dynamics from human demonstration for robot motion planning. In: ASME Dynamic Systems and Control Conference (DSCC)

Ravichandar H, Dani AP (2017) Human intention inference using expectation-maximization algorithm with online model learning. IEEE Transactions on Automation Science and Engineering 14(2):855–868

Ravichandar H, Kumar A, Dani A (2016) Bayesian human intention inference through multiple model filtering with gaze-based priors. In: 19th International Conference on Information Fusion (FUSION), pp 2296–2302

Ravichandar H, Salehi I, Dani A (2017) Learning partially contracting dynamical systems from demonstrations. In: Proceedings of the 1st Annual Conference on Robot Learning, PMLR, vol 78, pp 369–378

Razin YS, Pluckter K, Ueda J, Feigh K (2017) Predicting task intent from surface electromyography using layered hidden markov models. IEEE Robotics and Automation Letters 2(2):1180–1185

Recasens A, Khosla A, Vondrick C, Torralba A (2015) Where are they looking? In: Advances in Neural Information Processing Systems (NIPS)

Russakovsky O, Deng J, Su H, Krause J, Satheesh S, Ma S, Huang Z, Karpathy A, Khosla A, Bernstein M, et al (2015) Imagenet large scale visual recognition challenge. International Journal of Computer Vision 115(3):211–252

Schaal S (1999) Is imitation learning the route to humanoid robots? Trends in cognitive sciences 3(6):233–242

Simon MA (1982) Understanding human action: Social explanation and the vision of social science. SUNY Press

Song D, Kyriazis N, Oikonomidis I, Papazov C, Argyros A, Burschka D, Kragic D (2013) Predicting human intention in visual observations of hand/object interactions. In: 2013 IEEE International Conference on Robotics and Automation, IEEE, pp 1608–1615

Strabala KW, Lee MK, Dragan AD, Forlizzi JL, Srinivasa S, Cakmak M, Micelli V (2013) Towards seamless human-robot handovers. Journal of Human-Robot Interaction 2(1):112–132

Traver VJ, del Pobil AP, Perez-Francisco M (2000) Making service robots human-safe. In: IEEE/RSJ International Conference on Intelligent Robots and Systems., pp 696–701

Tsai CS, Hu JS, Tomizuka M (2014) Ensuring safety in human-robot coexistence environment. In: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp 4191–4196

Warrier RB, Devasia S (2017) Inferring intent for novice human-in-the-loop iterative learning control. IEEE Transactions on Control Systems Technology 25(5):1698–1710

Xiao J, Hays J, Ehinger KA, Oliva A, Torralba A (2010) Sun database: Large-scale scene recognition from abbey to zoo. In: Computer vision and pattern recognition (CVPR), 2010 IEEE conference on, IEEE, pp 3485–3492

Yao B, Jiang X, Khosla A, Lin A, Guibas L, Fei-Fei L (2011) Human action recognition by learning bases of action attributes and parts. In: Internation Conference on Computer Vision (ICCV), Barcelona, Spain.

Yarbus AL (1967) Eye movements during perception of complex objects. In: Eye movements and vision, Springer, pp 171–211

Zhou B, Lapedriza A, Xiao J, Torralba A, Oliva A (2014) Learning deep features for scene recognition using places database. In: Advances in Neural Information Processing Systems 27 (NIPS), pp 487–495