Intention Estimation with Recurrent Neural Networks for Mixed Reality Environments

Michael Fennel
Intelligent Sensor-Actuator-Systems Laboratory (ISAS)
Institute for Anthropomatics and Robotics
Karlsruhe Institute of Technology (KIT), Germany
michael.fennel@kit.edu

Serge Garbay
ARSPECTRA
Foetz, Luxembourg
serge.garbay@arspectra.com

Antonio Zea
Intelligent Sensor-Actuator-Systems Laboratory (ISAS)
Institute for Anthropomatics and Robotics
Karlsruhe Institute of Technology (KIT), Germany
antonio.zea@kit.edu

Uwe D. Hanebeck
Intelligent Sensor-Actuator-Systems Laboratory (ISAS)
Institute for Anthropomatics and Robotics
Karlsruhe Institute of Technology (KIT), Germany
uwe.hanebeck@kit.edu

***Abstract*—Knowledge about human intention can be beneficial in many disciplines of robotics, such as collaborative manufacturing, prosthetics, or encountered-type haptics. Existing intention estimation approaches are either traditional and rely on hand-crafted features and heuristics, or learning-based and tailored to very specific conditions. This paper attempts to combine the best of both worlds by making recurrent neural networks adaptable to different scenarios. To achieve this, the intention estimation problem is formulated as a probabilistic classification problem and two new data sets with real-world motion and eye-tracking data are presented. Based on this data, three real-time capable classifiers with different features regarding situational awareness and additional outputs are designed and evaluated against two competing approaches. The results show that two out of three classifiers lead to improved or equivalent performance compared to traditional approaches, while good generalization is maintained.**

***Index Terms*—intention estimation, intention recognition, dataset, recurrent neural network, LSTM, machine learning, mixed reality, XR, VR, AR**

I. Introduction

Collaborative robots have become an important part of state-of-the-art manufacturing processes. Although functional safety is already on a sufficient level for many applications, most measures are reactive, i.e., they minimize the collision impact or perform an evasive action [1], [2]. A promising approach to further minimize hazards in human-machine collaboration is the adoption of predictive methods that try to foresee the next human actions and proactively avoid corresponding workspace regions. To achieve this, a suitable estimation of the current human intention is required. Beyond collaborative robots, intelligent prosthetics that can adapt to the next object being grasped could benefit from knowledge about human intention [3]. Regarding extended reality (XR) applications, the estimation of the human intention serves as a core component for so-called redirected walking technologies [4], [5]. Furthermore, a dynamic scenario adaptation is possible if the user's intention is known, e.g., to avoid the user leaving the play area in the user environment.

Another use case for intention estimation is found in the context of encountered-type haptics [6]. Imagine a user walking within their user environment to a closed door in a virtual (target) environment. When the user reaches for the handle of the door to open it, they will actually reach out for the handle-shaped end effector of a very large kinesthetic haptic interface, such as [7], programmed to render the mechanical properties of the door. If multiple handle-like objects allowing interaction exist in the scene, either multiple haptic manipulators must be available, which is costly, or the haptic manipulator must be positioned at the next object the user is going to interact with. This, in turn, requires the estimation of the current human intention.

With these use cases in mind, the goal of this paper is to present a novel, wearable intention estimation system. In contrast to existing Bayesian or heuristic approaches, this system does not rely on hand-crafted features or an extensive set of hyperparameters due to the utilization of recurrent neural networks. Based on two new data sets with real-world data, we prove that the performance of our proposed estimators is on par or even better than existing systems.

II. Problem Statement

In the considered scenario as illustrated in Fig. 1, a single user can move around freely by natural locomotion in their real (user) environment, which is statically mapped to a virtual (target) environment. The user is equipped with a head-mounted display (HMD) providing information about the user's gaze direction $\underline{g}^H$ and the head pose H composed of translation $\underline{x}_{\mathrm{H}}^{\mathrm{W}}$ as well as rotation $\mathbf{C}_{\mathrm{H}}^{\mathrm{W}}$, whereas W denotes a fixed reference coordinate system. In the user's real or virtual environment, there is a fixed number $n < N$ of non-overlapping, point-like

This work was supported by the ROBDEKON project of the German Federal Ministry of Education and Research.

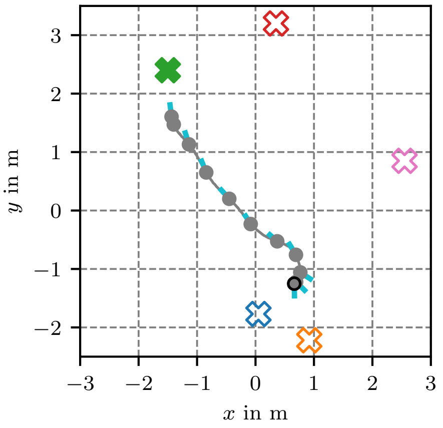


Fig. 1. 2D projection of the example scenario for intention estimation. The potential objects are marked with colored crosses and the trajectory of the user's head is depicted in gray, whereas the black circle locates the starting point. Gray circles mark samples that are $0.5\,\mathrm{s}$ apart and blue lines represent the gaze direction. In this example, the user pursues the solid green cross.

objects, where N is an upper bound for the number of objects. Each object with index i is fixed in space with the known 3D coordinates $\underline{x}_i^{\mathrm{W}}$.

With this information, a system shall be designed that estimates the user's intention $\underline{p} = \big(p_1, \ldots, p_n\big)$, where $0 \leq p_i \leq 1$ states the probability that the user is going to interact with object i next. Optionally, the predicted intention vector can be augmented with the value $p_{\varnothing}$ holding the probability that the user is going to interact with none of the known objects next. In both cases, the sum over $\underline{p}$ is 1. Beyond that, the system must be able to produce online estimates of the user intention, thus the resulting algorithm must be real-time capable, e.g., the frame rate must be $\geq 10\,\mathrm{Hz}$.

III. Related Work

The recognition of human intention is an active field of research, appearing in various domains, including autonomous driving, industrial robots, and neuroscience with a variety of different techniques [8]. Furthermore, intentions can be predicted on different abstraction levels. An example of a very low level is given in [9], where an intention recognition algorithm is presented to distinguish between a transportation and a positioning phase. At the other end of the spectrum, methods such as the one presented in [10] facilitate the prediction of an intended high-level task. To obtain the intention on an intermediate abstraction level (e.g., the user intends to grab object A), neuronal networks (NNs) and probabilistic algorithms are two groups of methods that are widely used [8].

A. Neural Networks

A recent noteworthy example from the first group is [11], where a Long Short-Term Memory (LSTM) NN is used to predict the object a user is going to grab next based on hand pose information. Similar to our initially outlined use case with encountered-type haptics, the intention information is then used for a technique called haptic retargeting. Despite this similarity, this method is not suitable for the problem in Section II as the targets are hard-coded and the hand, which is mostly not visible while walking, is the only considered feature. Similar approaches that use LSTMs for predicting the next object are [12] and [13]. Although these methods replace hand pose information with gaze measurements, they still have the limitation that the objects have predetermined positions and that the user is not allowed to move freely.

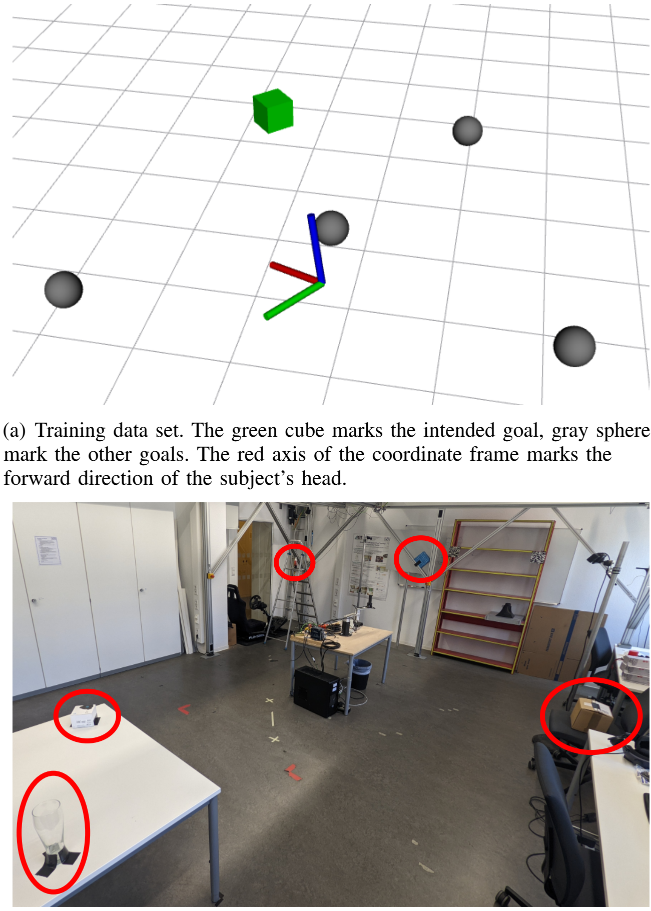
To the best of our knowledge, no NN-driven approach exists that is suitable to solve the stated intention estimation problem with a moving user and variable object positions. However, two probabilistic approaches were found in the literature. As they will be used as references in our evaluation, they will be explained briefly in the following.

B. Bayesian Estimator

In [14], the HMD instrumentation provides user position, walking direction and gaze information that is processed using a Wonham filter, which is a special recursive Bayesian estimator for discrete-valued states. To achieve this, a hidden Markov model (HMM) with a state for each possible object is deployed. The transition matrix, used in the prediction step, is handcrafted with a single tunable hyperparameter describing the probability of staying in that state. The measurements are not processed directly but converted to features. These are distance, relative gaze, and relative walking direction with respect to the goal. This way, the update step of the Wonham filter is independent of the actual object position, simplifying the measurement model. The necessary conditional measurement probability densities are modeled using Gaussian mixtures, which are fitted to real-world data. As a result, the probability for each potential goal is obtained. Other values, such as the probability for a goal that is unknown to the intention recognition, are not provided.

C. HAIR

Similar to the previous approach, the *head-mounted AR intention recognition* (HAIR) method utilizes an HMM to estimate the user's intention while moving around in a warehouse [15] or when standing in front of an industrial robot [16]. In addition to the hidden state for each potential object, two states describing an irrationally acting user and a user without intention are included, respectively. The transition matrix has four hyperparameters and, as before, is tuned manually. To incorporate measurement data from the head-mounted device, including object position, gaze, and head or hand position, a feature vector called motion validation vector is calculated: For each object, the inner product of gaze and line of sight to the object is multiplied with a hand-crafted quantity that becomes 1 when the hand or the head is moving towards an object and 0 when it is moving away from an object. This vector is then used to calculate a likelihood vector based on simple heuristics, which, in turn, is used as an emission matrix for the HMM. Eventually, a Viterbi algorithm is applied to predict the most probable state path in the HMM and therefore the user's current intention.



(a) Training data set. The green cube marks the intended goal, gray spheres mark the other goals. The red axis of the coordinate frame marks the forward direction of the subject's head.

(b) Evaluation data set. Five distinguishable items (white box, brown box, blue box, glass, and bottle) were placed at fixed positions in the room. The obstacle in the middle (table) is meant to produce more diverse trajectories.

Fig. 2. Scenarios that were used for capturing the new data sets.

While both approaches, the Bayesian estimator and HAIR, are proven to work, they rely on hand-crafted heuristics which might limit their performance and their ability to generalize. At the same time, current NN-powered approaches are not ready to be used with the given problem class. For this reason, a new intention estimator is needed.

IV. Data Set

A data set, that contains head pose, gaze, and goal positions together with the underlying human intention, is required to evaluate the performance of any estimator that solves the above-stated problem. Furthermore, the same kind of data is required to fit the Gaussian mixture in Section III-B and to train the proposed NN. While there exist some data sets that capture head pose and gaze data, such as OpenNEEDS [17], none of them contains information about the true user intention, which is crucial for training a classifier. For this reason, we decided to create two different time series data sets of approaches on our own using a *Microsoft Hololens v2* with a frame rate of 60 Hz. Both datasets are available online.[1]

A. Training Data Set

First, a data set for the training was created with seven different subjects. Each of the subjects first got a short briefing about the task, then the Hololens was introduced and calibrated. After a short trial period to get comfortable with the holographic projections, the subjects performed consecutive approaches to different goals for 10 minutes. To achieve this, five virtual point-like objects are randomly placed on a $4\,\mathrm{m} \times 4\,\mathrm{m}$ grid with $1\,\mathrm{m}$ resolution and a height that randomly varies up to $0.2\,\mathrm{m}$ around the participant's head height as seen in Fig. 2(a). For each approach, one object is randomly highlighted and the time on a dial clock is presented as a directional cue to the subject. While the subject approaches the given object, head pose, gaze data, and object positions are stored together with the highlighted object, which is considered to be identical to the subject's intention. As soon as the vicinity of the highlighted object is reached, the goal is removed, a new object is randomly spawned, and the process is repeated.

In a postprocessing step, temporarily missing gaze data is filled with a sample and hold strategy. Furthermore, implausible approaches (e.g., the subject fails to obey the given intention) are discarded by selecting all approaches with a duration above the 95 %-percentile. As a result, 763 out of 804 captured approaches are included in the data set, resulting in about one hour of time series in total.

B. Evaluation Data Set

While the previous data set contains good positional diversity regarding the 2D placement of the potential goals or intentions, it suffers from the very narrow field of view of the Hololens. This yields a reduced height-placement diversity and extensive, but unnatural seeking phases at the beginning of approaches, which impair supervised learning processes and an unbiased evaluation.

To overcome this issue and to prevent data leakage, another data set with ten subjects was created for evaluation purposes. For this data set, objects are now physically placed and fixed in the room as depicted in Fig. 2(b), which means that the initial seeking phase disappears once the subject is familiar with the scene. The preparation of the subjects remains the same, except that the initial warm-up phase is replaced with a room tour. As a result, the goal heights vary more and the Hololens is just used as a data recorder. During the capture process, which is now 5 minutes per subject, the desired goal is commanded by voice instead of HMD overlays to minimize visual distractions. Additionally, a table is placed in the middle of the scene to add more variety to the approaches. After applying the same post-processing procedure as in the previous data set, 559 out of 589 approaches with a total duration of 35 minutes remain. Fig. 1 shows an example from the resulting data set.

V. Intention Estimation with Recurrent Neural Networks

In the following, three network architectures and the relevant information about input encoding and training is presented for the given problem.

[1] https://doi.org/10.5281/zenodo.7687774

A. Input Encoding

To achieve real-time inference, a recurrent network architecture is used, meaning that the time series data is fed sample-wise into the network. All object positions $\underline{x}_i^{\mathrm{W}}$ are transformed into ego-centric coordinates $\underline{x}_i^{\mathrm{H}}$ based on the head pose information. As the user's gaze direction $\underline{g}^{\mathrm{H}}$ is already measured in head coordinates, the head pose can now be entirely omitted in the input data because the scene in front of the user is fully reconstructible at any time. As a result, redundant input data is removed. Furthermore, it simplifies the training of any NN because it does not need to learn the necessary transformations.

B. Architecture

As discussed in Section III-A and [18], LSTMs are a reasonable choice when it comes to the online processing of time series data. Considering that the input data is low-dimensional, standard LSTMs, as they are provided by TensorFlow, are used in the following network architecture.

1) Binary Classififer: According to [19], a set of binary classifiers can lead to better results for very specialized classification tasks than a jointly trained multi-class classifier. Since all measurement data is available in ego-centric coordinates, it is then obvious to replace a joint estimator for the intention probability vector $\underline{p}$ with an estimator for each object. Hence, a binary classifier is trained. As seen in Fig. 3(a), the network gets the gaze information and the relative pose of the object i, whose intention rating p_i shall be estimated. The information is processed in several cascaded LSTM layers before a fully connected layer with a sigmoid function calculates the one-dimensional classification result $\hat{y}$. Consequently, an output value of 1 means that the user intends to interact with the considered object, while 0 means that the user has no intention for object i.

For the training, positive examples are created by taking an approach to a goal and labeling it with 1. An equal amount of negative examples can be easily created by replacing the object's position data in an approach with data from another random object and labeling it with 0. To deploy the resulting classifier for inference, the sigmoid function in the fully-connected layer is removed first. Then, a stack of n binary classifiers with identical weights is connected to a softmax layer. This way, all binary outputs are normalized and combined to the joint intention $\underline{p}$.

2) Extended Binary Classififer: The presented binary classifier can be used to estimate the user's intention for a given set of objects. However, information describing the object's environment is not incorporated. To improve the resulting isolated behavior (e.g., in a scenario where two objects are in close vicinity), it is beneficial to include information about all objects in the input data. A possible realization of this idea is shown in Fig. 3(b), where $N-1$ auxiliary 3D inputs are added in comparison to Fig. 3(a). Each of these inputs provides the ego-centric position of one of the targets for which the binary classifier is not making a prediction. The remainder of the network architecture remains unaltered.

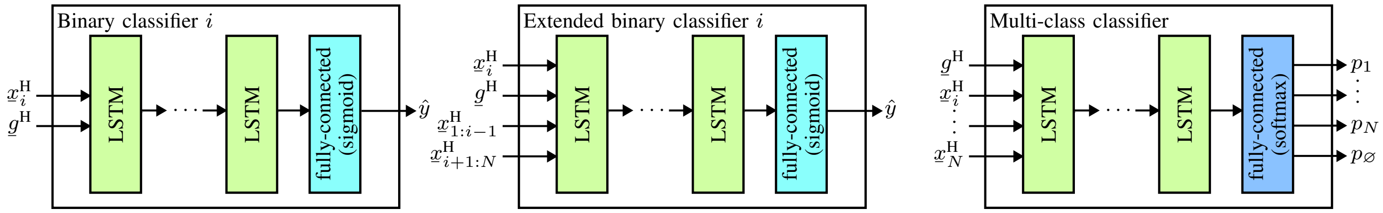
Classifier	# LSTM layer	# hidden states	Dropout
Binary	3	36	0.4
Extended binary	3	64	0.2
Multi-class	1	64	0.2

TABLE I
HYPERPARAMETERS OF EACH ARCHITECTURE

The resulting classifier can be deployed for $n \leq N$ goals by stacking n instances and replacing the sigmoid function with a softmax layer as before. If less than N objects are to be included, the position values of the additional unused inputs are set to a value that does not occur in real data sets, e.g., $(-10, -10, -10)$. During the training with the data set from Section IV-A, $n = N = 5$ holds. To learn the handling of unused inputs, the data set is augmented: With a probability of $5\,\%$, each auxiliary position input is set to the above-mentioned special value, while the remaining data of the example under consideration is kept. In addition, the auxiliary inputs are permuted randomly in each epoch.

3) Multi-Class Classififer: Although the extended binary classifier is able to incorporate all available information into its decision-making in theory, it still has no chance to output that a user has no intention or an intention not associated with a known goal. As stated in Section II, we summarize this as *no known target* with the probability $p_\varnothing$. This can be interpreted as an extra classification result. Thus, the introduction of an additional output is promising. Unfortunately, this is not feasible in the case of a binary classifier. Therefore, a multi-class classifier with the architecture as depicted in Fig. 3(c) is introduced. On the input side, the gaze vector and all object positions in ego-centric coordinates are stacked. The output $\hat{\underline{y}}$ is a $N+1$ element vector, where the first N elements quantify the probability of each object and the last entry equals $p_\varnothing$.

For the training, data augmentation steps are necessary. In the first step, the object numbering during each approach is permuted randomly. This way, the network is invariant towards which input supplies the object information matching the user's intention. As the number of object inputs is now fixed to N, but fewer objects might occur during inference, the input value of unused inputs can be set to a special value as done for the previous classifier. For this reason, each object is set to the special value with a probability of $5\,\%$ in each approach in a second augmentation step. If the object position of the labeled intention is deleted this way, the output label is adjusted to *no known target*. The third augmentation step is required for the last output because the training data itself does not contain the label *no known target*. To still train the classifier, a goal is removed randomly from the training data in each approach. The remaining input data is kept and assigned with the desired *no known target* label. As a consequence, the network can be trained for a maximum number of $N = 4$ objects for the given data set with five objects. If the *no known target* output is not included in the estimation, the third augmentation step can be omitted and $N = 5$ different intentions can be distinguished.


Binary classifier i
$\underline{x}_i^{\mathrm{H}}$
$\underline{g}^{\mathrm{H}}$
LSTM
LSTM
fully-connected (sigmoid)
$\hat{y}$
Extended binary classifier i
$\underline{x}_i^{\mathrm{H}}$
$\underline{g}^{\mathrm{H}}$
$\underline{x}_{1:i-1}^{\mathrm{H}}$
$\underline{x}_{i+1:N}^{\mathrm{H}}$
LSTM
LSTM
fully-connected (sigmoid)
$\hat{y}$
Multi-class classifier
$\underline{g}^{\mathrm{H}}$
$\underline{x}_i^{\mathrm{H}}$
$\underline{x}_N^{\mathrm{H}}$
LSTM
LSTM
fully-connected (softmax)
p_1
p_N
$p_\varnothing$

(a) Binary classifier during training.

(b) Extended binary classifier during training.

(c) Multi-class during training and inference.

Fig. 3. All proposed network architectures utilize LSTM layers as the core building block. Only the multi-class classifier can be used directly for inference. For the other classifiers, multiple instances need to be stacked and combined using a softmax layer for inference.

C. Training

For the training, the time series data from Section IV-A was split in a 70/20/10 ratio for training, validation, and test. As all subjects showed a seeking phase at the beginning of each approach, which is not typical for intrinsic intention, the first 0.5 s were removed from each training sample. To improve the performance, an automated hyperparameter tuning regarding the number of LSTM layers between 1 and 5, the number of hidden states between 8 and 64, and the dropout probability ranging from 0.2 to 0.4 was performed for the three different network architectures. Table II shows the final architectures for every network. The networks with the best hyperparameters were then retrained with relaxed stopping criteria. Since the approaches are varying in length and TensorFlow requires padded input data for implementation reasons, the custom cross-entropy loss function

$$L = -\frac{1}{N}\sum_{i=0}^{N}\sum_{t=0}^{T_i}\frac{\underline{y}_{it}^{\top}\log\left(\hat{\underline{y}}_{it}\right)}{T_i} \tag{1}$$

was created to ensure that each approach is weighted equally, independent of duration. Here, N is the number of examples and T_i is the number of samples in example i. The desired output vector of example i at time t is $\underline{y}_{it}$. The actual output vector is marked with a hat. Analogously, the custom accuracy function

$$A = \frac{1}{N}\sum_{i=0}^{N}\sum_{t=0}^{T_i}\frac{\delta\big(\mathrm{maxind}\big(\hat{\underline{y}}_{it}\big) - \mathrm{maxind}\big(\underline{y}_{it}\big)\big)}{T_i} \quad , \tag{2}$$

was defined. In this expression, $\mathrm{maxind}(\cdot)$ returns the index of the maximum element and $\delta(\cdot)$ is the discrete Dirac delta function.

VI. Evaluation

In the following, the proposed intention estimators are evaluated on the data sets from Section IV and some example situations are discussed. In addition, the special case of *no known target* is analyzed.

A. Estimator Performance

For comparison, the multi-class classifier from Section V-B3 was first trained without the *no known target* output, hence $N = 5$. Beyond that, all NNs were trained according to Section V-C. The resulting numbers of parameters are listed in the last column of Table II. For comparison with the Bayesian approach from Section III-B, a Gaussian mixture with 50 components was fitted on the training data of the NNs. Despite the high number of components, overfitting was not observed in combination with the validation data. In the HAIR method from Section III-C, the Viterbi-algorithm was replaced with a Wonham filter and the parameters from [15] were used because this combination resulted in the highest accuracy.

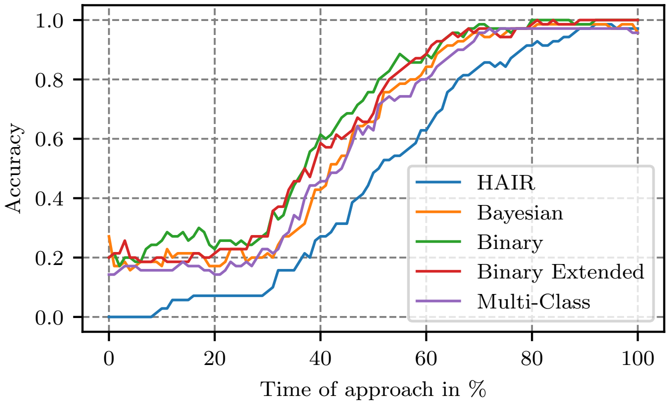


Fig. 4. Average estimation accuracy with respect to the relative time of an approach based on the test data in the data set from Section IV-A.

The first five rows of Table II separately quantify the performance on the test part of the training data set (*test data*) and the evaluation data set (*evaluation data*). In general, the evaluation data set yields higher accuracy and smaller performance variations among the different estimators, which is caused by the simpler but more realistic evaluation scenario. Although the accuracy measure is subject to noise due to the comparatively small data set, it can be seen that the binary and the binary extended classifier are outperforming the Bayesian approach on the test data. On the evaluation data, no outperformance is observed, but the binary extended classifier is on par with the Bayesian classifier. The HAIR approach is significantly worse in all scenarios and therefore will not be further discussed. In Fig. 4, the accuracy of all estimators is plotted with respect to the relative time progress of the approaches on the test data. During the first 20 % to 30 %, all estimators except for the binary classifier seem to be guessing.

Another perspective on the estimator performance can be obtained when T_{correct}, the average remaining time of an

	Training Data Set (Test Part)		Evaluation Data Set		Processing Time in ms	Number of NN parameters
	Accuracy in %	T_{correct} in s	Accuracy in %	T_{correct} in s		
HAIR	48	2.08	68	2.49	0.17	–
Bayes	61	2.28	78	2.62	0.45	–
Binary classifier	66	2.67	76	2.48	1.24	27253
Binary Extended classifier	64	2.58	78	2.73	1.12	87361
Multi-class classifier w/o *no known target*	59	2.39	75	2.60	0.66	21573
Multi-class classifier with *no known target*	58	2.16	72	2.34	0.53	20805

TABLE II

METRICS OF THE PROPOSED NN FOR INTENTION ESTIMATION ON TWO DIFFERENT DATA SETS.

approach, in which the predicted goal is correct and stable, is evaluated. The corresponding columns with T_{correct} in Table II reveal that the extended binary classifier is superior to the Bayesian classifier for both data sets. Interestingly, the T_{correct}-values of the multi-class classifier are better than or on par with the Bayesian classifier. This indicates, that it tends to produce more stable estimates than the Bayesian classifier at the end of an approach, despite its reduced accuracy.

The confusion matrices depicted in Fig. 5 provide insights into the information utilized by the NNs. Specifically, the first two columns/rows and the subsequent two align with adjacent objects in Fig. 2. This also explains the slightly increased confusion when one of these objects is the target.

All of the presented methods can be considered real-time capable with frequencies beyond $100\,\text{Hz}$ based on the average processing times in Table II, which were measured on an Intel Core i7-11800H CPU without utilizing a GPU. The binary classifiers take more time than the multi-class classifiers, as multiple binary instances must be calculated in each iteration.

B. Example Situations

With the (extended) binary classifier performing better than the Bayesian classifier, the multi-class classifier seems to be uninteresting based on the accuracy metrics. However, we found that the behavior of the different estimators is not entirely reflected by the selected metrics as illustrated in the exemplary approach from Fig. 6. Since the example is selected from the data set Section IV-A, it takes about $1.5\,\text{s}$ till the subject has understood the command and found the target to approach. Regardless of this seeking phase, the Bayesian estimator in Fig. 6(c) immediately strongly predicts the blue goal, which is not among the intention. After $1.5\,\text{s}$, the Bayesian estimator continues to predict the correct goal until the prediction is shifted abruptly to the pink goal and back to the red one after $2.0\,\text{s}$ and $2.3\,\text{s}$, respectively. This jumpy behavior is explained by the very particular likelihood regarding the relative motion to the target and the parameter-induced tendency of the Markov chain to change its state. In practice, this is very unsuitable for downstream systems trying to take advantage of the probability information. In contrast to that, the proposed intention estimators are much less certain about their predictions at the beginning.

Furthermore, it should be noted, that the binary classifier in Fig. 6(d) already predicts the right object very early, when the user is still looking towards the blue object. Although this is beneficial for high accuracy, it is probably not the behavior a human supervisor would expect in the given situation. Similarly, the result from the extended binary classifier in Fig. 6(e) seems to be erratic during the first $1.5\,\text{s}$. In contrast to that, the result from the multi-class classifier in Fig. 6(f) is more plausible as the probability for the red goal is increased, but not dominating at the beginning.

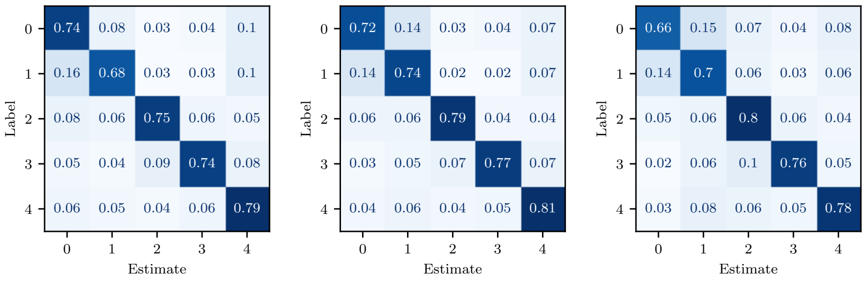
C. No Known Target

In contrast to the binary and extended binary classifier, the multi-class approach can handle a user without (known) intention, which was defined as *no known target* above. The last line of Table II indicate a performance drop regarding the accuracy and T_{correct}. Nevertheless, the representative example in Fig. 7 shows, that the intention estimation produces reasonable results: In the beginning, the user is facing the green object and not moving, hence the probabilities for green and *no known target* are highest. For a short period, the user is facing the red object while turning counterclockwise, making the system predict red as the most probable intention. Afterwards, the probability for *no known target* becomes highest as desired. Furthermore, the probabilities for the remaining objects in front of the user (i.e., orange and blue) are increased, which seems plausible as the user suddenly may get the intention to approach these.

In the example from Fig. 8, the same network is used when a user approaches a known goal. Interestingly, the network predicts *no known intention* at the beginning, although the corresponding initial seeking phase never has been labeled that way in the training data. This behavior can be considered a successful generalization because the seeking phases of the approaches contain no intrinsic intention and therefore might be interpreted as labeled incorrectly.

VII. CONCLUSIONS

From the proposed recurrent networks for intention estimation, the binary and the binary extended classifier have demonstrated their capability to outperform the existing algorithms regarding the examined quantitative measures on a challenging test data set. Further experiments on a separate, less challenging evaluation data set prove that both approaches are able to generalize, although the clear outperformance regarding a Bayesian approach is lost due to the simplicity of the scenario. Therefore, especially the extended binary classifier architecture seems to be a promising solution. The resulting benefits are that the positions of adjacent goals are automatically incorporated and that there is no longer a need for manual feature design



(a) Binary classifier. (b) Extended binary classifier. (c) Multi-class classifier.

Fig. 5. Confusion matrices of the proposed estimator architectures on the evaluation dataset.

as is the case for the Bayesian approach. From a quantitative point of view, the multi-class classifier failed to outperform the state-of-the-art. Nevertheless, the handling of *no known target* becomes feasible and the results look more plausible from a qualitative point of view. This raises the question of whether the chosen accuracy measure is the most suitable for the given problem or whether more appropriate metrics exist.

In future work, we will be dealing with improving the network architecture, especially regarding the multi-class classifier. Since all information is available to the multi-class NN, it should be possible to reach the same accuracy as the binary (extended) classifier in theory. In this context, we also consider the adoption of transformer-based architectures. Additionally, we intend to integrate the proposed intention estimation into our currently developed encountered-type haptic interface. In the long run, the inclusion of additional input data sources, such as hand pose, posture, and locomotion information is conceivable.

References

[1] S. Haddadin, A. De Luca, and A. Albu-Schäffer, “Robot collisions: A survey on detection, isolation, and identification,” *IEEE Transactions on Robotics*, vol. 33, no. 6, pp. 1292–1312, Oct. 2017.

[2] A. De Luca and F. Flacco, “Integrated control for pHRI: Collision avoidance, detection, reaction and collaboration,” in *2012 4th IEEE RAS & EMBS International Conference on Biomedical Robotics and Biomechatronics (BioRob)*, Jun. 2012, pp. 288–295.

[3] P. Weiner, J. Starke, S. Rader, F. Hundhausen, and T. Asfour, “Designing prosthetic hands with embodied intelligence: The KIT prosthetic hands,” *Frontiers in Neurorobotics*, vol. 16, 2022.

[4] N. C. Nilsson, T. Peck, G. Bruder, E. Hodgson, S. Serafin, M. Whitton, F. Steinicke, and E. S. Rosenberg, “15 years of research on redirected walking in immersive virtual environments,” *IEEE Computer Graphics and Applications*, vol. 38, no. 2, pp. 44–56, Mar. 2018.

[5] N. Nitzsche, U. D. Hanebeck, and G. Schmidt, “Motion compression for telepresent walking in large target environments,” *Presence: Teleoperators and Virtual Environments*, vol. 13, no. 1, pp. 44–60, Feb. 2004.

[6] V. R. Mercado, M. Marchal, and A. Lécuyer, ““Haptics on-demand”: A survey on encountered-type haptic displays,” *IEEE Transactions on Haptics*, vol. 14, no. 3, pp. 449–464, Jul. 2021.

[7] P. Rößler, T. Armstrong, O. Hessel, M. Mende, and U. Hanebeck, “A novel haptic interface for free locomotion in extended range telepresence scenarios,” in *Proceedings of the 3rd International Conference on Informatics in Control, Automation and Robotics (ICINCO 2006)*, Aug. 2006, pp. 148–153.

[8] M. Awais, M. Y. Saeed, M. S. A. Malik, M. Younas, and S. R. I. Asif, “Intention based comparative analysis of human-robot interaction,” *IEEE Access*, vol. 8, pp. 205 821–205 835, Nov. 2020.

[9] N. Stefanov, A. Peer, and M. Buss, “Online intention recognition for computer-assisted teleoperation,” in *2010 IEEE International Conference on Robotics and Automation*, May 2010, pp. 5334–5339.

[10] D. Gehrig, P. Krauthausen, L. Rybok, H. Kuehne, U. D. Hanebeck, T. Schultz, and R. Stiefelhagen, “Combined intention, activity, and motion recognition for a humanoid household robot,” in *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Dec. 2011, pp. 4819–4825.

[11] A. Clarence, J. Knibbe, M. Cordeil, and M. Wybrow, “Unscripted retargeting: Reach prediction for haptic retargeting in virtual reality,” in *2021 IEEE Virtual Reality and 3D User Interfaces (VR)*, Mar. 2021, pp. 150–159.

[12] C. Gomez Cubero and M. Rehm, “Intention recognition in human robot interaction based on eye tracking,” in *Human-Computer Interaction – INTERACT 2021*, Aug. 2021, pp. 428–437.

[13] J. Pettersson and P. Falkman, “Human movement direction classification using virtual reality and eye tracking,” *Procedia Manufacturing*, vol. 51, pp. 95–102, Jan. 2020.

[14] P. Rößler, O. C. Schrempf, and U. D. Hanebeck, “Stochastic prediction of waypoints for extended-range telepresence applications,” in *2nd International Workshop on Human Centered Robotic Systems (HCRS 2006)*, 2006.

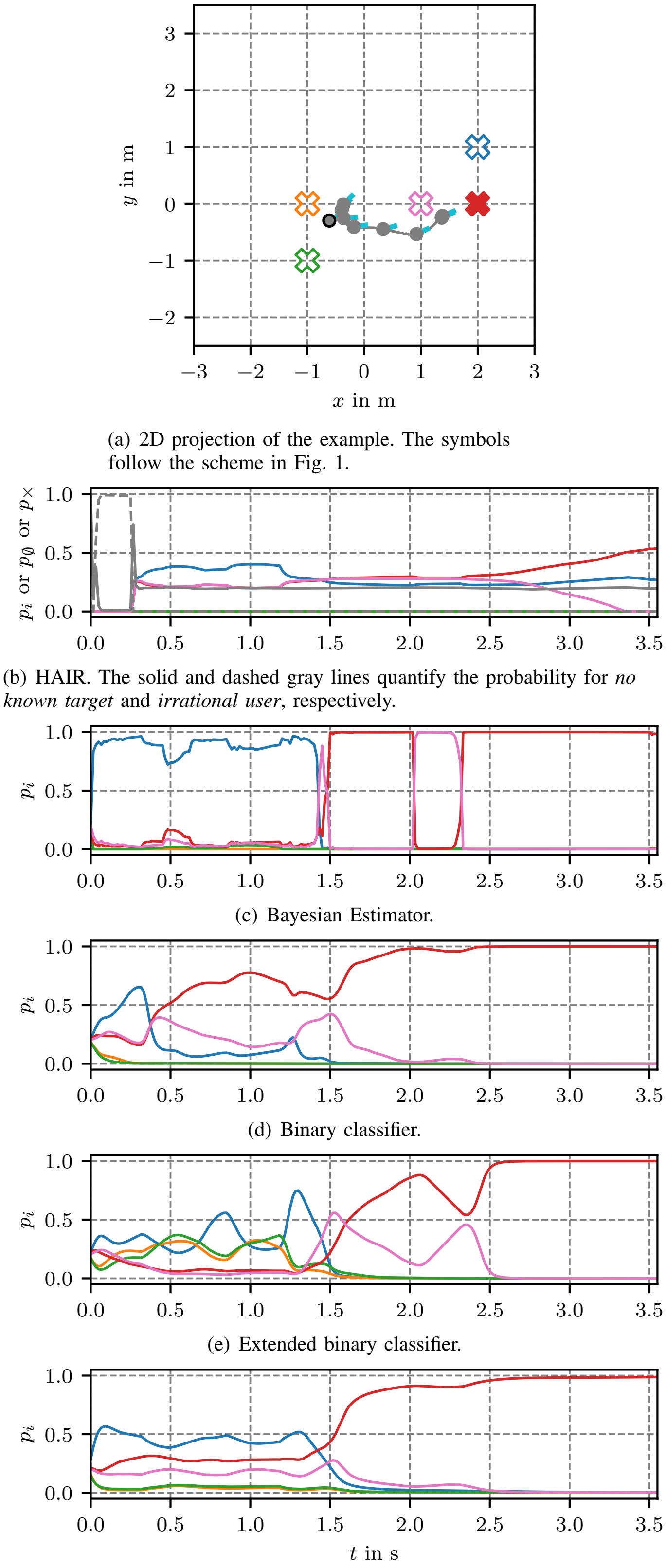
[15] T. Petković, D. Puljiz, I. Marković, and B. Hein, “Human intention estimation based on hidden markov model motion validation for safe flexible robotized warehouses,” *Robotics and Computer-Integrated Manufacturing*, vol. 57, pp. 182–196, Jun. 2019.

[16] D. Puljiz, B. Zhou, K. Ma, and B. Hein, “HAIR: Head-mounted AR Intention Recognition,” in *4th International Workshop on Virtual, Augmented, and Mixed Reality for HRI*, Mar. 2021.

[17] K. Emery, M. Zannoli, J. Warren, L. Xiao, and S. Talathi, “OpenNEEDS: A dataset of gaze, head, hand, and scene signals during exploration in open-ended VR environments,” in *ETRA ’21 Short Papers: ACM Symposium on Eye Tracking Research and Applications*, May 2021.

[18] B. Lindemann, T. Müller, H. Vietz, N. Jazdi, and M. Weyrich, “A survey on long short-term memory networks for time series prediction,” *Procedia CIRP*, vol. 99, pp. 650–655, 2021.

[19] T. J. D. Berstad, M. Riegler, H. Espeland, T. de Lange, P. H. Smedsrud, K. Pogorelov, H. K. Stensland, and P. Halvorsen, “Tradeoffs using binary and multiclass neural network classification for medical multidisease detection,” in *2018 IEEE International Symposium on Multimedia (ISM)*, Dec. 2018, pp. 1–8.



(a) 2D projection of the example. The symbols follow the scheme in Fig. 1.

(b) HAIR. The solid and dashed gray lines quantify the probability for *no known target* and *irrational user*, respectively.

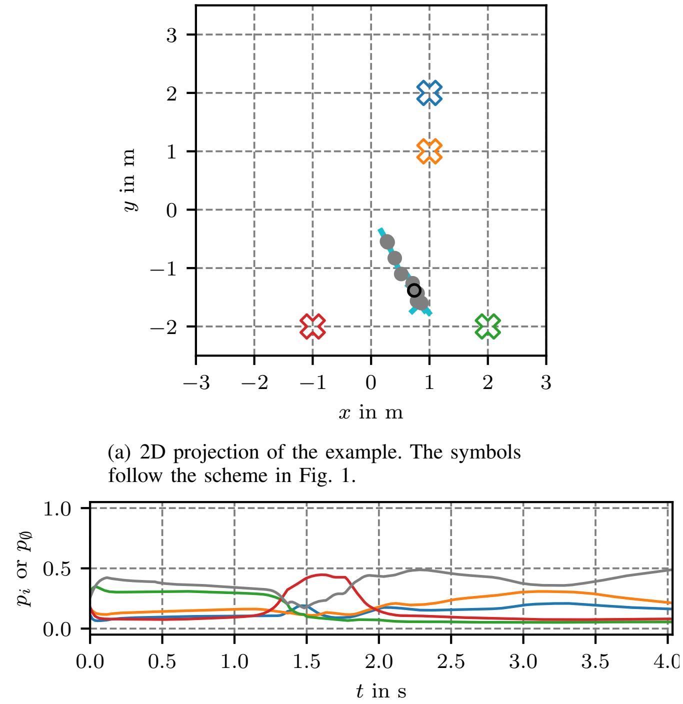
(c) Bayesian Estimator.

(d) Binary classifier.

(e) Extended binary classifier.

(f) Multi-class classifier.

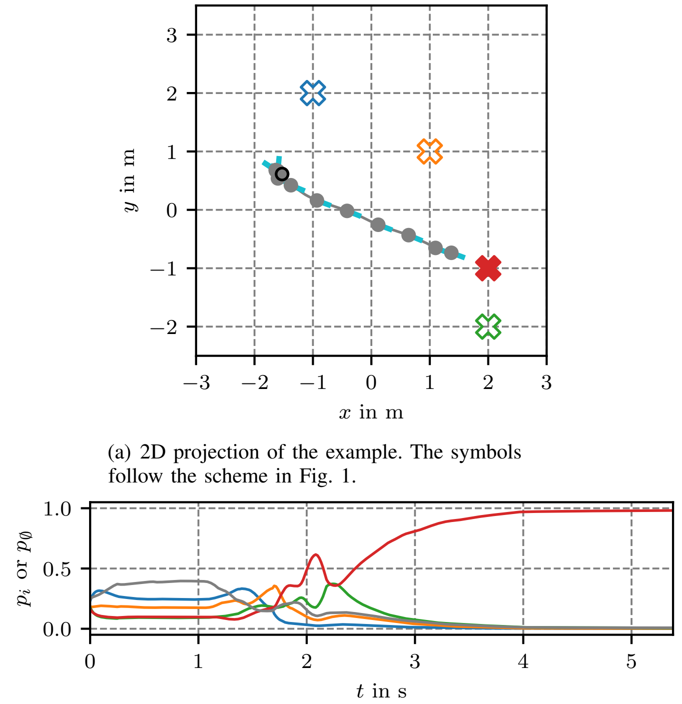
Fig. 6. The behavior of all estimators in an exemplary scenario, where the user pursues the red object. Here, the multi-class classifier was trained without the *no known target* output.



(a) 2D projection of the example. The symbols follow the scheme in Fig. 1.

(b) Predictions of the multi-class classifier. The grey line quantifies the probability of *no known intention* $p_\varnothing$.

Fig. 7. The behavior of the multi-class classifier in an exemplary scenario where the user follows an intention that is not among the known targets.



(a) 2D projection of the example. The symbols follow the scheme in Fig. 1.

(b) Predictions of the multi-class classifier. The grey line quantifies the probability of *no known intention* $p_\varnothing$.

Fig. 8. The behavior of the multi-class classifier in an exemplary scenario where the user has an initial seeking phase causing the multi-class network to predict *no known intention*.