Institute of Parallel and Distributed Systems

University of Stuttgart
Universitätsstraße 38
D–70569 Stuttgart

Masterarbeit

# Neural network user intent prediction for robot teleoperation

Hangbeom Kim

| | |
|---|---|
| **Course of Study:** | Infotech |
| **Examiner:** | Prof. Dr. rer. nat. Toussaint |
| **Supervisor:** | Ph.D. Jim Mainprice,<br>M.Eng. Yoojin Oh |
| **Commenced:** | June 11, 2018 |
| **Completed:** | December 11, 2018 |

## Abstract

The focus of this thesis is to design a machine learning framework that predicts user intent during robot teleoperation using eye and hand gesture tracking devices. In this thesis, we focus on a natural pick-and-place task. A simple virtual environment simulates the operation of a gripper as well as collects input data from hand gestures and eye movements.

Teleoperation technology allows to remotely operate robotic systems located in hostile, and inaccessible environments. Such environments are often incompletely known, and therefore not suitable for fully autonomous robots to operate in. In these situations, teleoperation can be used, which combines the problem-solving capabilities of the human with the precision and durability of the robot. Robots are extremely good at performing accurate precision motions, while humans are proficient in making complex planning and decisions.

Recent systems succeed by first identifying their operator's intentions, typically by analyzing the user's direct input through a joystick or a keyboard. Further, to enhance this method, the user input is combined with semi-autonomous control in shared autonomy. Especially, the indirect inputs like eye gaze data are a plentiful source of information for assessing operator intention. When people perform manipulation tasks, their gaze center tends to observe goal objects before starting the movements towards the corresponding objects and also glimpse the objects during the tasks.

In this thesis, we present an intent predictor and we compare it to six different models, developed using two low dimensional time series inputs (hand motion and eye movements tracking), to predict user intent. These models trained hand motion tracking, eye gaze tracking, and a combination of hand tracking and eye gaze tracking, respectively. With this implicit information, the models based on long short-term memory (LSTM) architecture for recurrent neural networks sequentially learn the goal region of the manipulation task regarding users intent. LSTMs are both general and effective at capturing long-term temporal dependencies.

From this study, the eye gaze and hand-based prediction model enables to understand the salient region faster and more accurate. The mean value and the standard deviation of the distance error between the reference goal position and the predicted position with the highest probability showed to be less than one cell in the 28x28 grid. Also, the probability distribution around the goal position in the reference data and the predicted data showed similar shapes with KL divergence of 0.4. Our findings underline the intent predictor for achieving efficient human-robot interaction works.

# Contents

# List of Figures

# List of Tables

# List of Abbreviations

**LSTM** Long Short-Term Memory. 3

**POMDP** Partially Observable Markov Decision Process. 13

**SVM** Support Vector Machine. 14

**FPS** Frame Per Second. 19

**MSE** Mean Squared Error. 22

**OpenGL** Open Graphics Library. 25

**PID control** Proportional, Integral, and Derivative control. 28

**KL divergence** Kullback-Leibler divergence. 34

# 1 Introduction

## 1.1 Motivation

We anticipate that robots can be deployed in environments where it is inaccessible or dangerous to humans, for example, disaster environments, underwater, or outer space. Robots can perform tasks such as reconnaissance of the environment and manipulation tasks with a great amount of force that exceeds the human ability [SK16]. However, despite the rapid advance in robotic technologies, it was shown at the 2015 DARPA Robotics Challenge that current robotic systems still lack in robustness and were unable to achieve full autonomy [POH17]. Human assistance is still required even for the robots to perform basic tasks, thus the performance of the robots partially depends on the ability of humans to teleoperate and interact with the robot [YNO+15].

Teleoperating a robot from a remote distance can be challenging. During teleoperation, the user relies solely on the robot's sensor data to perceive the robot's surrounding environment, mainly using the camera mounted on the robot. However, due to the limited field of view of the camera, the user experiences a "keyhole effect" as if he is looking through a keyhole [WTFR04] thus worsening the user's situation awareness. In addition, the user has to control the robot with interfaces which are primitive and unintuitive [SK16]. This can lead to human operator errors which can lead to task failure or even unrecoverable actions of the robot. It was discovered that during the 2015 DARPA Robotics Challenge, operator errors took a large fraction of the total errors [Tea15].

By sharing control between the semi-autonomous agent and the user, the robot's performance can be improved. Rather than the robot executing what the user commands, the agent predicts what the user wants and assists the user's commands [DS13]. In shared autonomy, predictions of user intent is a key challenge [JBS15; RLD18]. Anticipating human activities enable an assistive robot to plan ahead for reactive responses, which can lead to increased performance of the robot. [KS16; SMF14].

## 1.2 Related work

**Shared autonomy**
Shared autonomy is a combination of teleoperation and automated assistance. Shared autonomy is divided into two main parts: predict the user's intent, and assistance for the intent in many prior works [AEK05; JBS15; KWLV05]. As the robot agent does not know the user's goal a priority, the task is defined as a Partially Observable Markov Decision Process (POMDP) with uncertainty over user's intent [JBS15]. Javandi et. al. [JBS15] solved the problem using hindsight optimization. Reddy et al [RLD18] used a model-free deep reinforcement learning algorithm to assist the user's commands to generate optimal action, with a separate recurrent LSTM network to predict the goal when it is not known.

**User Intent Prediction**

Shared autonomy algorithms improved assistive robot control systems and reduced the amount of control by a user. In shared autonomy, predicting the user's goal is significant to accurately assist robots as we mentioned above. In state-of-the-art shared autonomy, their methods rely only on handling with a continuous direct input (e.g. joystick, mouse) [JBS15]. Further, the user intent of future actions is additionally appeared by the indirect signals like the gaze. Shared autonomy systems can exploit this implicit information when the users perform their task.

Analyzing eye movement has been researched and increasingly became popular in the domain of human-computer interfaces since 1947 [OMSC08]. This is because eye gaze data can be applied in multiple tasks, such as activity recognition [BWGT11] and understanding of mental states like cognitive load [ASK+18], by combining with machine learning algorithms. People's intentions and next actions are revealed by their eye gaze. In a natural task, eye gaze data typically leads to manipulation data (hand data) for targets with an abrupt start. Detecting a salient region is studied by many works to predict people's attention with neurobiological approaches [BI13]. Specifically, in pick-and-place tasks, when picking up an object under circumstances, the gaze was mainly fixed on the object until the hand is about to touch [PHL01]. With this user behaviour, probability distributions with POMDP over goal states are suggested to predict user intent [AS16]. Moreover, to understand expected actions from the user, Support Vector Machine (SVM) based classifier model is designed and it shows 76 percent accurate in predicting the customers' intended requests with the frequency of glances in a sandwich-making scenario [HASM15; HM16].

However, SVM tends to be sensitive to the data source and shows relatively low accuracy of 76 percent. Also the SVM model constrained by the specific flow and context of the interaction. Therefore, there are limitations in achieving generalizable to a wide range of contexts [HASM15; HM16]. For that reason, we suggest our intent predictor using the LSTM in order to overcome these challenges in previous works.

**Long Short-term Memory**

Recurrent neural networks (RNNs) are a powerful model for sequential data. RNNs efficiently learn implicit relationships between sequence elements. Image captioning, speech synthesis, and music generation all require that a model produce sequence outputs [Lip15]. LSTMs are a subset of RNNs that are able to deal with remembering information for much longer periods of time. LSTMs lead to not only many more successful runs, but learn also much faster compared to other conventional RNN models. Moreover, LSTMs deal with the vanishing gradient problems, which is a huge difficulty in training artificial neural networks. This is all achieved by using three gates (input, output, and forget gates) of LSTMs [HS97].

With the comparative characteristics of LSTMs, the LSTMs are dealt with by numerous studies for prediction through time series data like forecast, speech recognition, word prediction, and so on [CMG+14; GMH13; SRR18; ZE16]. And it has shown recent success on speech recognition [GMH13] and natural language tasks such as machine translation [CMG+14]. Therefore, we propose using LSTM model to train the time series data for eye gaze and hand movements.

## 1.3 Approach

In the thesis, we extend previous works related to shared autonomy with eye gaze and hand movements. Our main contribution is building the hand and gaze-based model for user intent prediction that can be integrated into the shared autonomy model.

The thesis is organized as follows. In Section 2, the approach of the research for the intent prediction model is briefly introduced. Then, based on this approach, we design a system architecture and the intent prediction model. Section 3 discusses experimental setups to explain our user study. Experimental results are presented to demonstrate the accuracy of our model in Section 4. In Section 5, discussions regarding limitation, user behaviours, and future works are stated based on our findings. In Section 6, the thesis is concluded.

# 2 Neural network user intent prediction

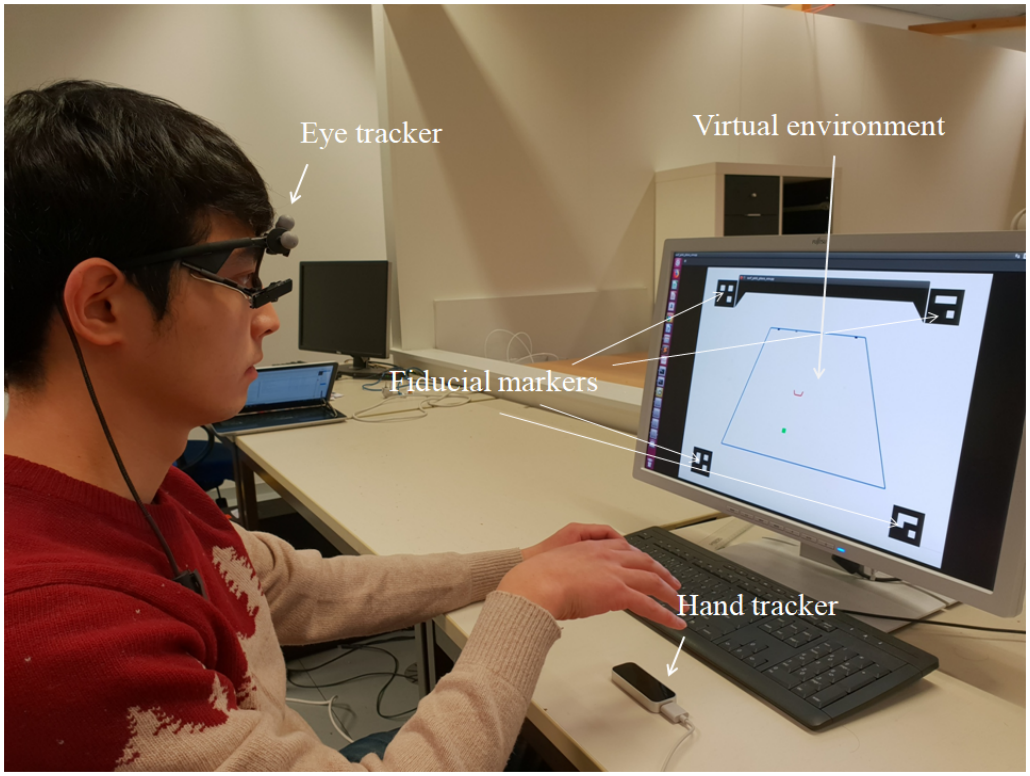



**Figure 2.1:** Intent prediction system based on an eye tracker and a hand tracker and a virtual environment for manipulation tasks

Our goal is to predict user intent in manipulation task simulation, especially in pick-and-place task, here. Our task is devised to pick up a gray object among four objects in the virtual environment.

Figure 2.1 shows the whole environment and our input devices. For sensing human's intent, we executed the task with two notable devices: an eye tracker and a hand gesture tracker. The gripper in the environment is manipulated by the hand tracker which retains abundant user intent. In addition, the eye gaze behaviours is also montiored by the eye tracker during the task.

Besides, in contrary to these previous studies regarding human-robot cooperation, the support system presented in this thesis uses the LSTM model to provide quicker and precise predictions with eye gaze and hand movement information for teleoperation. We train the model with trajectories of 15-time steps. The 15-time steps were chosen to give the best empirical results. In the next section,

we show that our prediction model achieved robust results against outliers and different behaviour styles from multiple users. Lastly, we compared six models with different training data sets to attain greater performances.

This chapter gives an overview of the system architecture and detailed algorithms. A brief of overview of the common models used to describe how data is accumulated and which devices are exploited. Furthermore, the details of the algorithm of the system are illustrated, along with a brief presentation of the diagram flow. Finally, the intent prediction model, which is one of the principal factors of the architecture, is concisely introduced in the last section.

## 2.1 System architecture

- Input :
  Mapped gaze point into the 2D plots (Pupil Labs)
  Velocity of hand movements (Leap Motion)
- Output:
  Predicted heat maps for goal positions

To enable a robot to understand user's aim, we propose a human intent prediction model that involves monitoring the points where the users are looking at and the behaviours how they react to grab a target object. The diagram below illustrates how information flows between different components of the overall system. Our system is made up of three main sections: **sense**, **perception**, and **act**.

Firstly, in the sensing part, eye gaze points and hand movements data are collected by two processes: Pupil Labs and Leap Motion. Pupil Labs provides an eye gaze tracking platform and it delivers the gaze information to an intent predictor model. Also, the user's hand movements are captured by the Leap Motion controller and send to our perception section. Secondly, our trained perception model is aware of user's targets by reading the inputs. The prediction model is designed by LSTM architecture and transposed convolution matrix. Finally, the robot would recognize the position of goal objects by the user and execute to pick the goal object up and place it in another location.

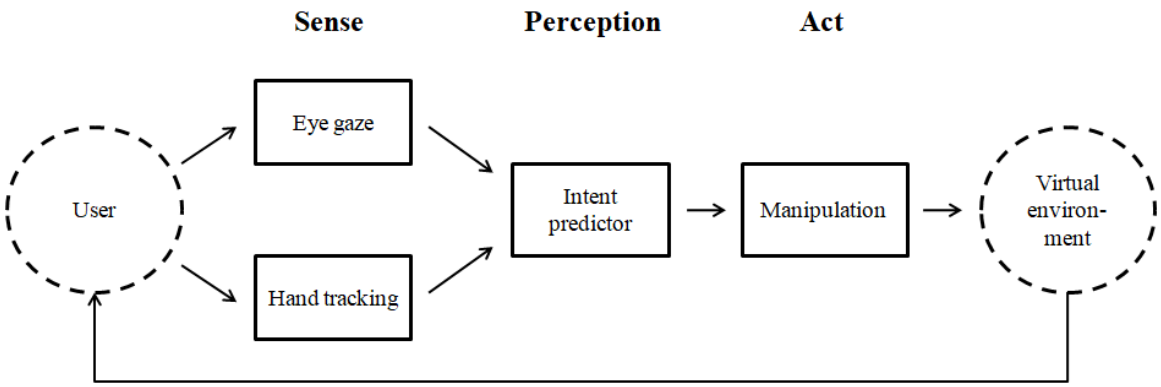



**Figure 2.2:** An overview of the system architecture

## 2.2 Algorithm

Time series data for gaze G and hand H are defined as an ordered sequence of n real-valued variables. By observing an underlying system over time, G and H are orderly collected. Elements of G stands for mapped gaze points (e.g. $g_1$ consists of $g_{1x}$ and $g_{1y}$) by the Pupil Labs software. H is also an abbreviation of the collected data of hand velocities in x-axis and y-axis.

$$G = \{g_1, g_2, ..., g_n\}, g_i \in R$$
$$H = \{h_1, h_2, ..., h_n\}, h_i \in R$$

Gaze data are transformed by Pupil labs software, defined as G', owing to the perspective view of the environment.

$$G' = \{g'_1, g'_2, ..., g'_n\}, g'_i \in R$$

Both input devices (an eye gaze tracker and a hand tracker) are sampled at 60 frames per second (FPS) to collect data. Thus, we can accumulate the data around 200 to 400 data points in each episode in the environment.

For communication between the hardware input devices (i.e., gaze and hand tracker) and the pick-and-place environments, we use ZeroMQ messages. ZeroMQ is a very lightweight messaging system. It is specifically designed for high throughput and low latency scenarios. ZeroMQ provides embeddable socket library that redefines the term socket as a general transport endpoint for atomic messages [Hin13]. ZeroMQ library was used to publish the needed information as inputs from an eye gaze tracker and a hand tracker. Also, the sent messages are subscribed by ZeroMQ in the virtual environment.

An intent predictor using LSTM based neural networks was trained with the collected data (G and H) and generates a trained model which can perceive user's intent in real time. We will discuss the detail of the LSTM architecture in the following section (2.3) since it is a significant factor to predict user intent. This model generates 28 by 28 probability heat maps to display the probability distribution of predicted goal position.

Using the heat maps from the eye gaze and hand data, our system recognizes users intent and smoothly control the gripper through PD controller (3.2.1).

$$H' = \{h'_1, h'_2, ..., h'_n\}, h'_i \in R$$

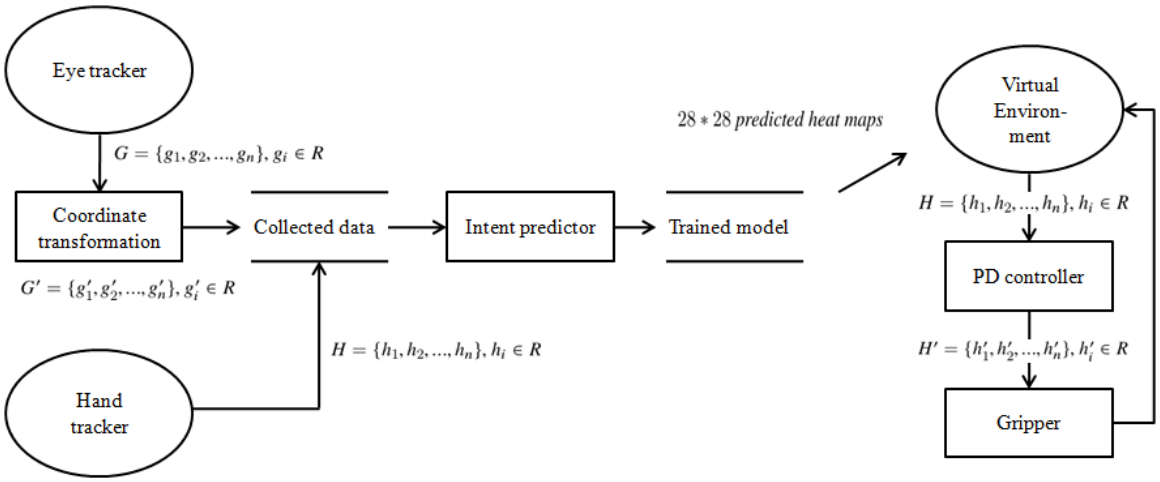



**Figure 2.3:** Data flow diagram of the system

## 2.3 Intent predictor model

### 2.3.1 LSTM model

LSTMs have potential to learn long-term dependencies. LSTMs can memorize information for long periods of time thanks to their inner cells which can carry information without changes. LSTMs have the form of a chain repeating modules of neural networks like all recurrent neural networks, but the repeating module is differently structured. The networks have the complete control over the cell state (the horizontal line running through the middle of the Figure 2.4), it can add, modify, or eliminate information in the cell using structures, called gates.

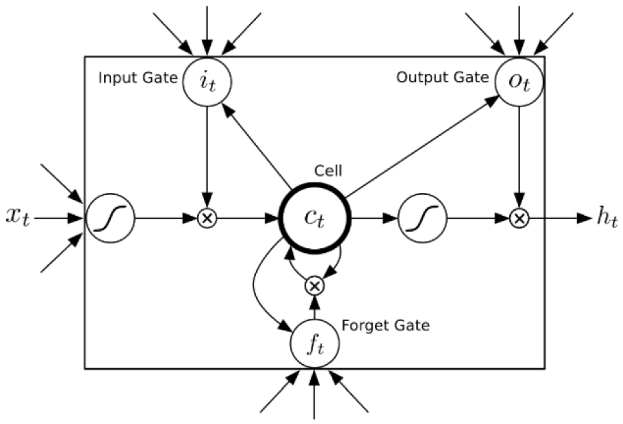



**Figure 2.4:** An LSTM memory block with input, output, and forget gates [GMH13]

Figure 2.4 illustrates that every memory blocks of LSTM includes three adaptive gating units: input, output and forget gate. Firstly, following equation shows forget gate layers with the sigmoid function decide what information is going to be kept or eliminated from the cell state.

$$f_t = \sigma(W_{xf}x_t + W_{hf}h_{t-1} + W_{cf}c_{t-1} + b_f) \tag{2.1}$$

Next, the selection of data to be updated is decided by multiplication of input gate layer with the sigmoid function and a tanh layer in (2.2).

$$i_t = \sigma(W_{xi}x_t + W_{hi}h_{t-1} + W_{ci}c_{t-1} + b_i) \tag{2.2}$$

To update the new cell state $c_t$ from $c_{t-1}$, old state $(c_{t-1})$ is multiplied by $f_t$, then $i_t * tanh$ is added in (2.3).

$$c_t = f_t c_{t-1} + i_t tanh(W_{xc}x_t + W_{hc}h_{t-1} + b_c) \tag{2.3}$$

We would drop the useless information and add the new practical data in this step. In the end, a sigmoid layer takes which parts of the cell state would be used. Then, the cell state goes through *tanh* and multiplies it by the output of the sigmoid gate (2.4, 2.5).

$$o_t = \sigma(W_{xo}x_t + W_{ho}h_{t-1} + W_{co}c_t + b_o) \tag{2.4}$$

$$h_t = o_t tanh(c_t) \tag{2.5}$$

### 2.3.2 Model Architecture

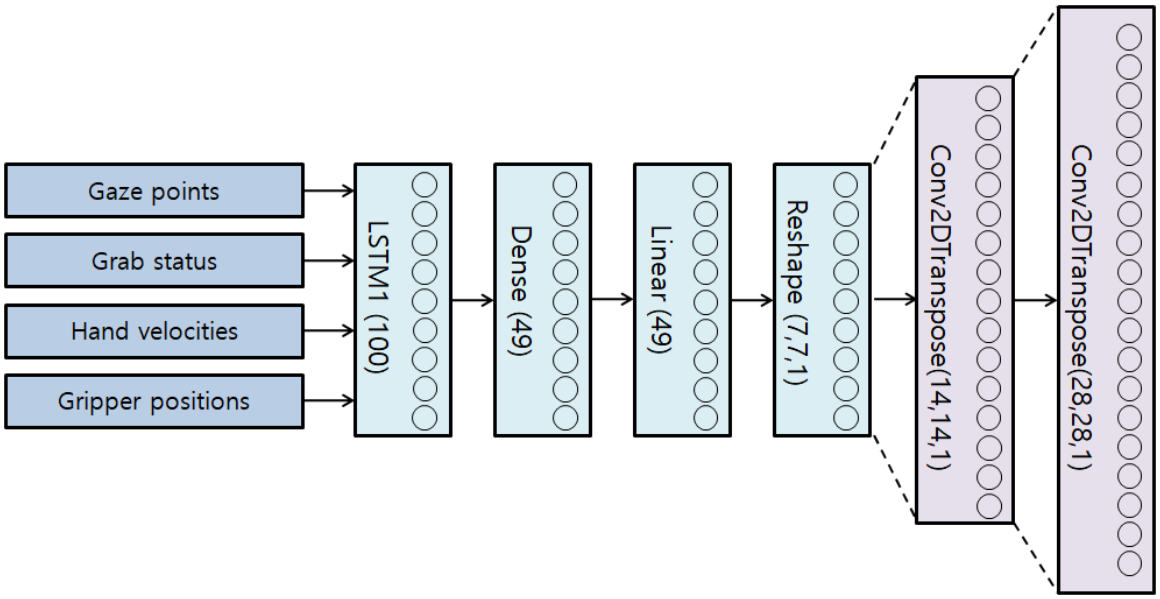



**Figure 2.5:** Intent prediction model architecture

- **Input:**
  Mapped gaze point into the 2D plots (Pupil Labs)
  Grab status
  The velocity of hand movements (Leap Motion)
  Current gripper positions on the environment
- **Output:**
  Predicted heat maps for goal positions

Figure 2.5 describes the architecture of our intent prediction model. Four different types of time series data above are inserted into the prediction model. Gaze points and hand velocities are major factors to analyze and predict operator's behaviour. The prediction model is trained by the trajectories of 15 data points in the past to prevent glimpsing obstacles and blinking eyes, and to understand the trends of user behaviours. And, we also feed grab status to convey the information on the different target to the model. This is because our target is converted from a target object to a goal place, as soon as the operator grabs the object. Furthermore, using the current positions of gripper gives the model the information on standard points.

We predict the salient region which includes the user's intent through the time series data. In our experiments, the LSTM model, trained by 2D points of object position in output space (a label), showed limitations to predict one specific point in abundant cases of a test set. These can be reasonably approximated by probability distributions. Therefore, the data type of the current object position is converted from the 2D points to its normal (Gaussian) distributions, before train the LSTM model. The probability density function of a continuous random variable can be derived from the distance between the object location and the center point of defined grid points (28 by 28 in our model). The normal distribution can be calculated using the probability density function.

**Normal distributions**
The normal (Gaussian) distribution is a continuous probability distribution. Real-valued random variables can be represented as the probability distribution using the normal distribution [Nad05].

The probability density of the normal distribution is

$$f(x|\mu,\sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \tag{2.6}$$

In our model, a set of distances between gaze points G and grid points is said to have the normal distribution with parameters $\mu$ and $\sigma$. The calculated distributions are inserted in our intent prediction model as a label.

A multilayer model consisting of the LSTM layer with 100 neurons, a dense layer, and two transposed convolution layers was implemented to form the base architecture for the model. A value of 100 hidden neurons has been chosen to train based on multiple experiments. In the dense layer, a linear activation function is applied to train the model. As a loss estimator, the mean squared error estimator (MSE), which is one of the most widely adapted ones, is used for the network:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(Y_i^p - Y_i^r)^2 \tag{2.7}$$

The latent matrices are reshaped into 7 by 7 matrices then they are enlarged by transposed convolution layers with two strides in each layer. Finally, two transposed convolution layers result in 28 by 28 neat heat maps for prediction of the goal position. We applied a supervised learning prediction method to obtain optimal parameters of the intent prediction model. Therefore, the predicted heat maps are consequently computed by the loss estimator with the original heat maps which contains the position of the target object.

# 3 Data collection setup

In this section, we describe the designs of an environment and how the users participate in experiments. As we mentioned in the introduction part, goal of our manipulation task was that picking up a gray object and place it in another position. First of all, general information on equipments of the system is introduced to present input data types. Secondly, experimental task design by Open Graphics Library (OpenGL) and tuned conditions in gaze and hand data is briefly explained to approach accurate predictions. Lastly, the information on participants and procedure for the experiments is briefly described.

The goal of the thesis is to predict the user intent during manipulation tasks, especially pick-and-place. A teleoperation environment was set up comprising a visual display of the robot's perspective, hand gesture sensor to command user input, and an eye tracker to track the user's gaze during teleoperation. To collect user's behavior during teleoperation, a teleoperation environment was set up in with a virtual view of the robot's task space.

## 3.1 Experimental task design and conditions

### 3.1.1 Environment design

To simulate a teleoperation scenario, a virtual environment is developed by OpenGL. OpenGL is a software interface to graphics hardware and it allows people to create interactive programs that produce color images of moving three-dimensional objects. The environment consists of multiple square objects, a goal position, and a gripper to simulate the robot's end effector, as shown in Figure 3.1. Among multiple objects colored black, the target object is gray, and the objective of the teleoperation task is to reach for the target object, pick up, then retrieve the object to the green goal position. The color of the target object is made similar to the other objects so that the operator pays attention to where the object is, otherwise the operator can detect the target object with just a quick glimpse after multiple turns of data collection. The virtual environment is laid in a tilted angle to simulate perspective view as if the camera is mounted on the head of a humanoid-like robot.

The operators asked to pick a gray object up through handling with the gripper. The object is automatically grabbed by the gripper in case of nearly approaching to the object. Then, the next step was moving the grasped object to the position of the green box. The scenario ended as soon as the gripper reached to the goal position to place the object.

Four Aruco markers were located on the edges of the virtual environment as reference points. The fiducial markers were able to do mapping of the position of our environment. Based on the mapping position of our environment, we drew a position where the operators glimpse at.

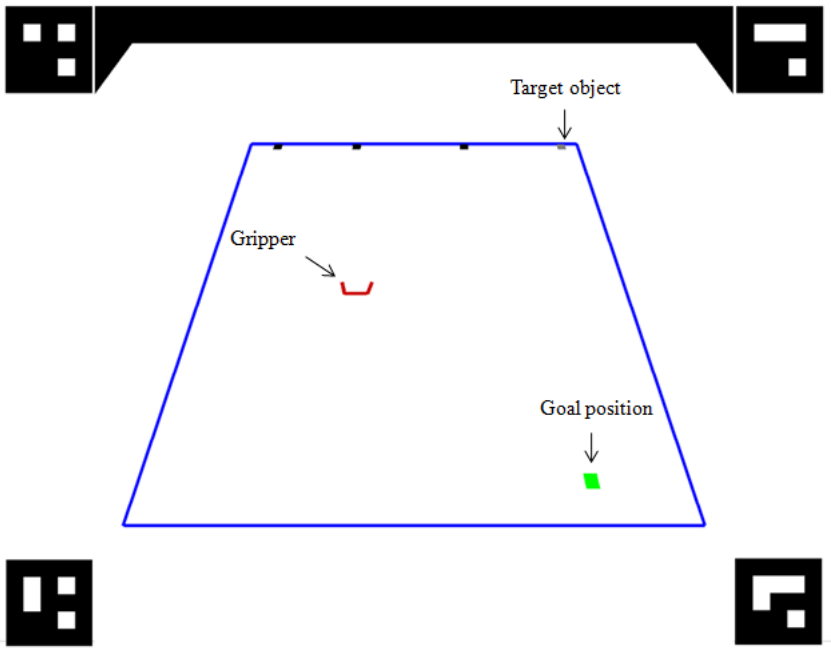



**Figure 3.1:** The setup of the virtual environment for pick-and-place tasks

### 3.1.2 Perspective projection transformation

In OpenGL, the most straightforward way of applying a projective coordinate transformation of the original experiment environment is to consider it to be a transformation to be applied to the projection matrix used in OpenGL [FTM02]. We were able to apply translations or rotations to change the default orientation of the viewing volume created by *gluPerspective()* utility function. The function *gluPerspective()* provides a way of defining frustum that is intuitive and convenient. The vertical angle that the viewport subtends at the eye in degrees is defined in the first argument of the *gluPerspective()* function. This angle is called $\alpha$ in Figure 3.2. The $\alpha$ must be positive and less than 180 degree. The height of the window in Figure 3.2 is $h = 2 * near * \tan\frac{\alpha}{2}$. The second argument is the aspect ratio (width/height) of the image, called *ar* [Figure 3.2]. This is the width of the viewport divided by its height. Thus, $w = ar * h = 2 * ar * near * \tan\frac{\alpha}{2}$. The nearest and furthest visible points, *near* and *far* in Figure 3.2, are determined by the last two arguments, respectively. With *gluPerspective()*, we needed to pick appropriate values for the field of a model-view transformation that simulates viewing ("looking at") the scene from a particular viewpoint, otherwise, the image looked distorted or showed a black screen. To solve this problem, *gluLookat()* function was available to define a viewing transformation. Our environment was designed by combining *gluPerspective()* and *gluLookat()* functions to be displayed as natural environments [Gro03].

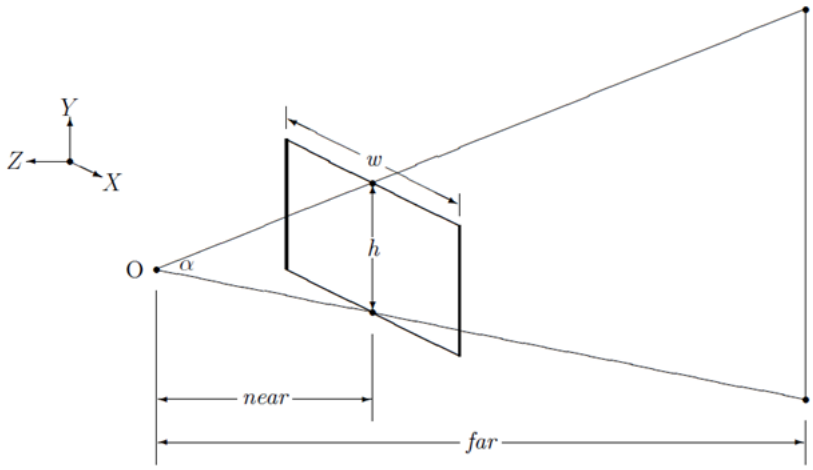



**Figure 3.2:** The perspective projection transformation

## 3.2 Equipments

### 3.2.1 Gripper contoller using Leap motion via PD control

In the teleoperation setup, the gripper is controlled by Leap Motion, which allows users to input gesture commands into an application in place of a mouse or keyboard. This hand tracker creates the potential for developing a general gesture recognition system in 3D domain [MYB15]. Full degrees of freedom for every moving part of the hand is provided by the hand tracker. The device can be easily set up and simply manipulated by novices compared to other motion trackers.

Leap Motion is a hand gesture and position tracking system with declared submillimeter accuracy [WBRF13].Leap Motion controller specifically captures the movements of a human hand using similar IR camera technology and IR light from LEDs. It tracks the hand in a space within a hemisphere of 1m in diameter above the sensor up to 200 FPS [MYB15]. The sensor is able to collect data such as fingertip positions, palm position and velocity, pinch strength, etc.

In the thesis, hand palm velocities data are collected from users to handle a gripper in a virtual environment and analysis user's behaviour. The Leap Motion controller provides 3D control, but two-axis control (only x-axis and y-axis of the hand velocities) are considered for our dataset because we propose the environment in a perspective projection view.

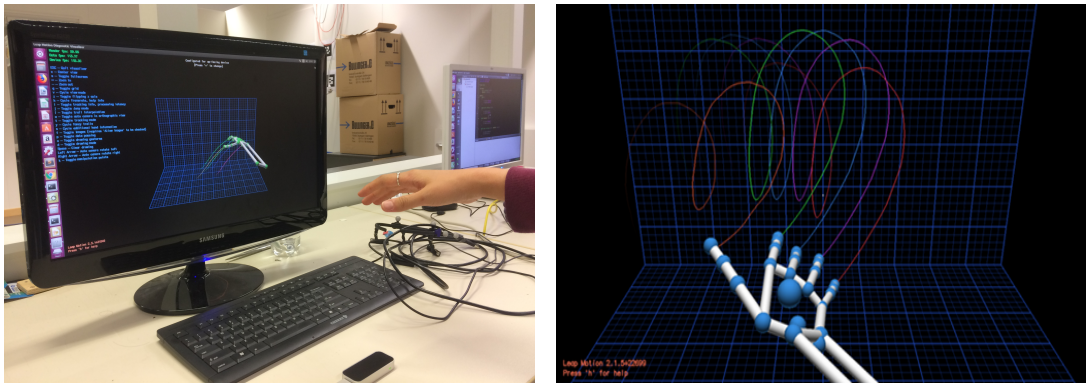


**Figure 3.3:** Pictures of operations and Visualizer through Leap Motion [MYB15]

**PID control**

There were difficulties in directly applying the velocities of hand palm from Leap Motion to the movement of the gripper due to its sensitiveness. Therefore, we utilized Proportional Integral and Derivative (PID control) control for regulation of speed of the gripper.

PID controllers are the most common way of using feedback in natural systems. PID controllers have been used in industrial control applications for ages long. PID controller is literally the output sum of three terms, proportional, integral, and derivative. The ideal version of the PID controller is given by the formula below [BSR12].

$$u(t) = k_p e(t) + k_i \int_0^t e(\tau)d\tau + k_d \frac{de}{dt} \tag{3.1}$$

where u is the control signal and e is the control error ($e = r–y$). The control signal is thus a sum of three terms: a proportional term: a proportional term that is proportional to error, and a derivative term that is proportional to the derivative of the error. The parameters are proportional gain $k_p$, integral gain $k_i$, and derivative gain $k_d$, respectively.

We applied a PD controller system to control the gripper in order to overcome its sensitivities. The control error was the velocity of hand movement in our system, and then, $\frac{de}{dt}$ was calculated by the deviation between the previous velocity and the current velocity. The aim of the PD controller is to enhance the stability of the system by improving control since it has the ability to prevent the following error of the system response. The derivative is taken from the output response of the system variable instead of the error signal, in order to overcome effects of the sudden change in the value of the error signal. Therefore, D mode is designed to be proportional to the change of the output variable to prevent steep changes occurring in the control output resulting from the steep changes in the error signal. In addition, D-only control is not utilized since D mode directly amplifies process noise.

### 3.2.2 Gaze point in projection view

The user intent can be additionally learned by an indirect input in the simulation environment. A head-mounted eye tracker provided by Pupi Labs is utilized to analysis eye movements. The ability to track and monitor eye behaviours is huge resources since people reveal their intentions and next actions through their eye gaze [BI13].

Pupil is an open source platform for pervasive eye tracking and mobile gaze-based interaction by Pupil Labs in Berlin. The Pupil comprises the open source software frame work, an accessible hardware device for a head-mounted eye tracking platform, and a graphical user interface to playback and to visualize video and gaze data [KPB14]. The eye tracker provides pupillary dilation in millimetre at a frequency of up to 60 Hz (120 Hz for pupil detection camera), as well as world camera footage displaying the current field of view of the operator in 720p resolution [GHF16].

To identify eye gaze and environment locations, calibration test is required. Then, based on the center positions of a pupil and a user-specific calibration routine, the gaze point is mapped to the viewed scene [FTBK16]. An average gaze estimation accuracy of 0.6 degree of visual angle (0.08 degree precision) can be provided by Pupil [KPB14]. An accurate calibration test is required for a robust tracking in natural environments. The location of pupil detection cameras and the focus of the camera lenses are crucial to successfully complete the calibration.

**Figure 3.4:** Pictures of wearing an eye tracker and the eye-camera view by Pupil Labs

**Projection view**
A world camera of the eye tracker detects four Aruco markers to track the positions of edges of the environment in real time. By reading the positions of Aruco markers, which can be seen in Figure 3.1, our system can adapt in the cases that users move their heads around or look at the environment at different angles. For this reason, Aruco marker tracking library from Pupil Labs was called to generate Aruco markers and track them.

The projection view of the task had to be considered into the point of gaze since the environment is realized in perspective view [Figure 3.5]. The projection view of the environment was defined by setting surfaces in Pupil platform. It was possible to set the range of the surfaces by dragging four edges of them using one or more fiducial markers. The surfaces conveniently defined with more than two Aruco markers were scanned by the tracking library even if some markers go outside the field of vision or are obscured. The registered surfaces were saved automatically and appeared when operators run Pupil platform in the future.


**Figure 3.5:** The window of Surface trackers using Pupil Labs

Moreover, the surfaces provide the calculation of the projection view of them through OpenCV libraries. In the surface trackers, a map matrix for a perspective transform can be calculated by four pairs of the corresponding points with the cv2.getperspectivetransform(src, dst) function (src indicates for coordinates of quadrangle vertices in the source image, and dst is in coordinates of the corresponding quadrangle vertices in the destination image).

$$\begin{bmatrix} t_i x_i' \\ t_i y_i' \\ t_i \end{bmatrix} = mapmatrix * \begin{bmatrix} x_i \\ y_i \\ 1 \end{bmatrix} \tag{3.2}$$

, where $(x_i', y_i') = dst(i)$ and $(x_i, y_i) = src(i)$, $(i = 0, 1, 2, 3)$

With the map matrix, we can apply a perspective transformations to an image. Here, Pupil utilized the cv2.warpperspective() function to describe the perspective projection view [Jos15].

$$dst(x, y) = src\left(\frac{M_{11}x + M_{12}y + M_{13}}{M_{31}x + M_{32}y + M_{33}} \quad , \frac{M_{21}x + M_{22}y + M_{23}}{M_{31}x + M_{32}y + M_{33}}\right) \tag{3.3}$$

, where $M_{ij}$ is the map matrix, $(i, j = 1, 2, 3)$

Finally, the gaze G' can also be computed by the function from original gaze G in Section (2.2).

## 3.3 Procedure

Twenty participants were recruited from the university. The age of participants (15 males and 5 females) ranged from 19 to 35 years. 15 percent of participants were bespectacled and the other 85 percent did not wear glasses. Data from one participant, who wore glasses, was excluded from the dataset due to a large error in the calibration test. The study took 20 minutes, and small rewards were given, afterwards.

An instruction sheet which included the objective of the teleoperation task and a short manual was provided to the participants before the study. In the beginning, the eye tracker was individually calibrated for each participant to align pupil positions from eye camera to scene space. The calibration was done using the calibration tool provided by Pupil Labs. The participant was then instructed on how to control the gripper using the hand gesture tracker and were given about 2 minutes of practice to reduce unfamiliarity before collecting the data. The data collection for each participant comprised of 2 trials, of which lasted 4 minutes each.

The participant was asked to sit down in a straight-right position, approximately 60cm to 100cm from the screen. The participant was asked to put on the head-mounted eye tracker. The position of and the focus of eye cameras were adjusted to make sure that the participant's pupils were properly tracked. The participant followed a 9-point screen marker based on a calibration routine in full-screen mode during the calibration test. Participants were advised to not move their head and not excessively blink their eyes during the calibration as it can cause calibration errors. The calibration test was repeated until ideal conditions (0.6 degrees of accuracy and 0.08 degrees of precision) were met based on [KPB14]. When controlling the gripper, the participant was asked to keep the hand at a certain height, 10 cm to 20cm above the Leap Motion to prevent loss of track. During the 2 minutes of practice, the gaze data was carefully monitored to check if all data was generated properly.

# 4 Experimental results

The total collected dataset consisted of 443,699 data points from 1476 episodes. Each episode lasted around 5 seconds and around 300 data points were collected from each episode, resulting in 20,000 to 25,000 data points for each participant during the whole experiment. A few data points were excluded from the dataset that deviated too far away from the environment range (0 to 1) due to hardware limitations.

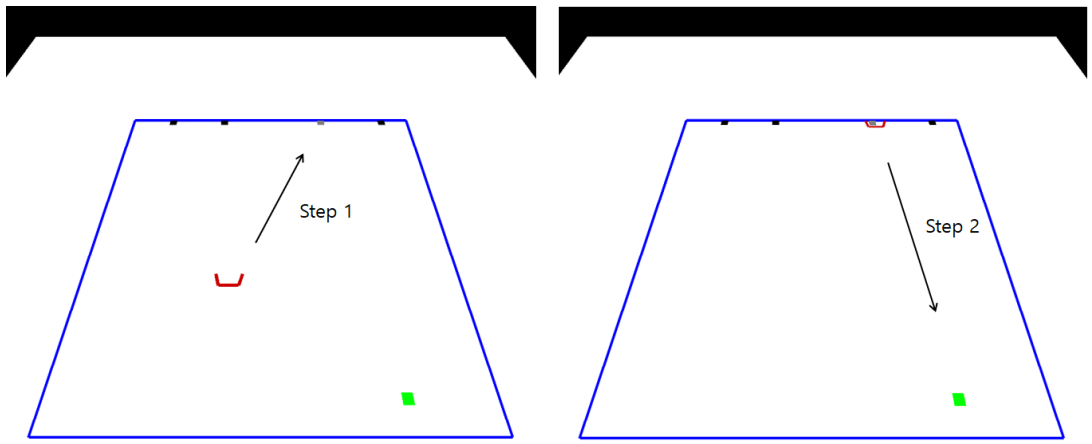
The dataset was split into two sets, 67 percent of the dataset was used to train the model and the rest was used to test the model. The model was used to train three different input cases: using only hand data, using only gaze data, and using both hand and gaze data, as mentioned in Section 2. In order to compare and analyze the characteristics of differences between one user and multiple users, we additionally designed other three different models. We defined the model and it can be seen in Table 4.1 below.

**Table 4.1:** Six different models through various data type

| | Gaze-only | Hand-only | Both |
|---|---|---|---|
| All participants | M1 | M2 | M3 |
| One participant | M4 | M5 | M6 |

Using the data we collected (Section 3), we first drew heat maps in test episodes to analysis general trends of gaze behaviours. Furthermore, we compared our prediction model results to the model, which is trained by one user, to infer the differences between hand movements and gaze behaviours information and to prove how robust and powerful our neural network model is.

## 4.1 Results




**(a)** 1st step **(b)** 2nd step

**Figure 4.1:** Two phases of the simulation

Each episode of the simulation has two steps as shown in Figure 4.1. The participant was asked to first go towards the object and then place the object at the green spot. Here, the first step is to grab the object (4.1a), then the second step retrieves the object to the green goal position as shown in Figure 4.1b. In the first step, the goal (the target of the human intent) is the object and in the second step the goal becomes the green goal position. We inserted the grab status into the intent prediction model in order to distinguish the two steps, which have different goals.

Figure 4.3 illustrates our predicted heat maps based on probability distributions. Compared to the target heat map (4.3a), which indicated for the position of the object, our prediction model showed significantly similar results (4.3b). In the different number of participants, the model trained by the mixture of hand and gaze data (M3 and M6) always showed better performances to predict the critical region in the virtual environment. This model, in general, was able to foresee the user intent in under one second (60 frames). Due to the different size of frames in one episode, the graph had to be downsampled in Figure 4.5. The trend of the graph described that the loss went down until the user grasped the object, then the increase of the loss was appeared by changing of the target position from the object to the goal position. Later, the loss continuously decreased during the task, again. The results of distance errors and Kullback-Leibler Divergence (KL divergence) supported to prove the efficient and accurate model. There was only one cell of distance error in 28 by 28 grid systems as well as the predicted probability distributions produced very similar heat map to the original target heat map.

## 4.2 Analysis

We predicted the salient region within the blue boundary line in Figure 4.1. The coordinates of probability distributions and gaze points corresponded to the square blue boundary region as if we were looking at the environment from a top view rather than the perspective view, as shown in Figure 4.2. Also, the coordinates were flipped vertically so that the objects lie near the x-axis.

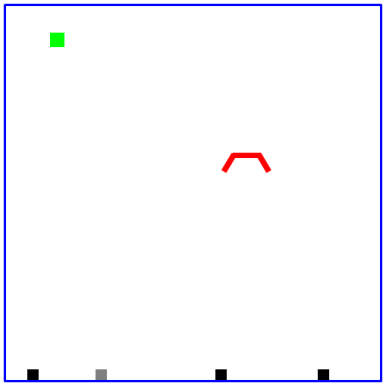


**Figure 4.2:** The environment shown in top view

We applied 15-time steps of gaze points to our intent predictor, the number of time steps being selected after multiple trials. The reason for this was because the gaze data often included data points that were irrelevant in predicting the goal. For instance, gaze data while looking at other objects or blinking led to noisy gaze data. Moreover, due to the hardware issues, the eye gaze points were not completely stable, but they were vibrated around the goal object.

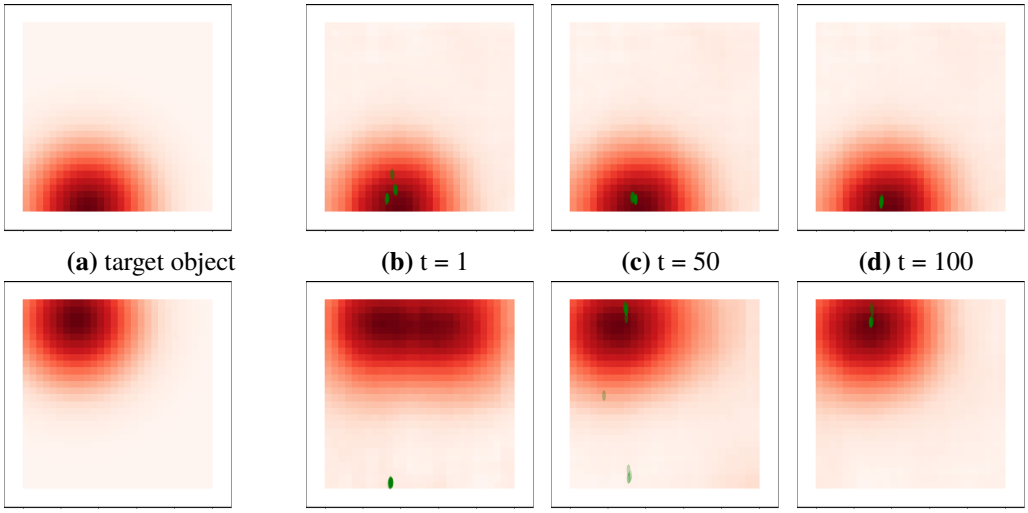
### 4.2.1 Each participant's gaze behaviours during the manipulation task

By analyzing the participant's data, we recognized the persistent patterns of gaze behaviours during the task. In general, the types of behaviours were divided into two groups, which were target-oriented people and gripper followers. Firstly, the gaze behaviours of target-oriented people were focusing on the object during the manipulation task. It was intuitively understandable to predict the user's intent since the positions of gaze point exactly corresponded to the goal object.

The results of one participant by our intent predictor were described as 28 by 28 heat maps in Figure 4.3. The graph in the most left hand side was the heat maps for showing positions of our targets. Next, the other graphs were placed by discrete time series in every 50 frames. And, every fifteen gaze points were plotted as green points in the graphs below. The brighter green colors stand for the more latest data.

Our model showed great predictions as soon as the pick-and-place task was executed by a user since the environment was designed to start the task when the user was ready and pressed the 'S' button on a keyboard. Our predicted probability distributions were stable until the gripper grasped the target object. Interestingly, when the target position was changed from the object to the goal location for placing the object in the graph (4.3f), our intent predictor model showed multimodal

probability distributions. We believed that this was because there was a lack of data to predict the goal location in our model. However, with the following gaze trajectories in (4.3g, 4.3h), the model precisely understood where the goal location was.

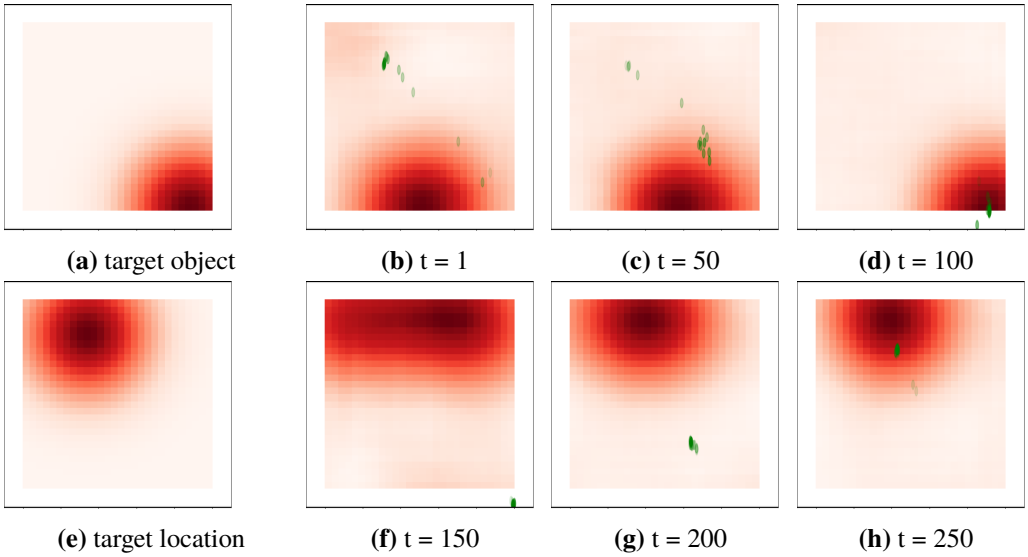


**(a)** target object **(b)** t = 1 **(c)** t = 50 **(d)** t = 100

**(e)** target location **(f)** t = 150 **(g)** t = 200 **(h)** t = 250

**Figure 4.3:** Target oriented user behaviours

On the other hand, gripper followers showed different behaviours. Gripper followers tended to look toward the gripper while they were moving it to the goal object [Figure 4.4]. They glimpsed at the target at first, and then they started to continuously follow the gripper. At first, the intent predictor underwent troubles on finding the target object. However, in around one second (50 to 100 frames in 60 FPS), the location of the target was able to be foreseen by our model via the trajectories of gaze points.



**(a)** target object **(b)** t = 1 **(c)** t = 50 **(d)** t = 100

**(e)** target location **(f)** t = 150 **(g)** t = 200 **(h)** t = 250

**Figure 4.4:** Gripper follower behaviours

### 4.2.2 Overall results

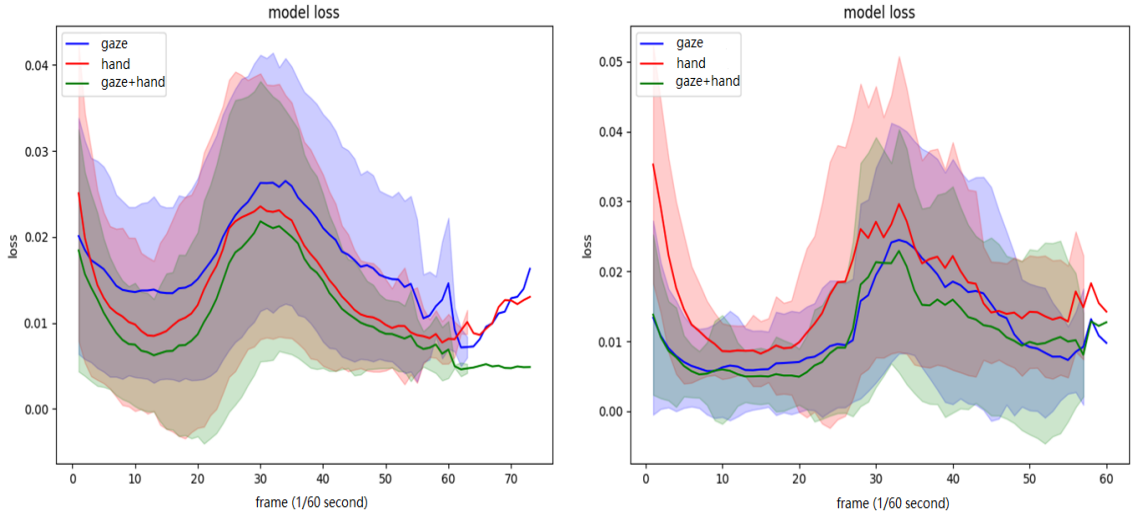
**Losses**
Figure 4.5 showed the mean and standard deviation of stacked mean squared error of probability distributions for the six models above in each episode. Due to the differences of length of time in each episode, downsampling was done to see a general tendency of the losses in the episode.

When it comes to models with all participants, the mean and standard deviation of three models (M1, M2, M3) were depicted in the red, green, and blue in 4.5a, respectively. The basic trends of the graphs were similar to each other. The losses of all the predictions were decreased until the users grabbed the target object at when the number of frames was around 10 to 15 (without downsampling, the real frames were 100 to 150, which meant 1.6 seconds to 2.5 seconds in 60 FPS). Then, as soon as the user picked up the object, the losses were steeply increased due to the change of the positions of targets. Lastly, the number of losses was continuously diminished, again, while the user placed the object in our goal position.

Mostly the model, trained by both gaze and hand data (M3), produced the lowest loss among the three models. In particular, hand gesture data (M2) had more influences on M3 compared to gaze behaviours (M1), as it could be seen by the mean and standard deviation in the graph 4.5a. Because each user had different gaze behaviours during the task, the mean of accumulated gaze data was higher than hand data. Moreover, many noises were able to be existed in gaze data due to hardware limitations or getting out of the calibrated range.

To analyze the results, we tried to evaluate the models within participants. The general trends of the two graphs were almost same, however, the models by one participant's data showed different aspects in hand-only (M5) and gaze-only data (M4) in the graph 4.5b. Most of the mean values of M4 were smaller than M5. The performances of M6 still remained in the highest rank. In particular, the amount of losses of M5 never reach the M6.

To sum up, we found significant differences in six models of when the participants manipulated the pick-and-place task. Between them, the intent prediction models with the mixture of both data (M3 and M6) resulted in the greatest performance of predictions, always. In single participant models, gaze-only data was still the dominant factors to forecast user intents when we experimented the test with five different participants. That proved how gaze source was useful and helpful to predict the user intent. On the other hand, as we collected more data, the hand data was more suitable to feed data in the intent predictor model than gaze behaviours. This was because there were various gaze behaviours from multiple users.




**(a)** between participants **(b)** single participant

**Figure 4.5:** Mean and standard deviation of losses with three different models

### Distances and KL divergence

To briefly investigate our models, we computed the distance errors and the KL divergence compared to original probability distributions. The distance errors were the length between the most probable position of the predicted probability distributions and the original location. Figure 4.6 described the mean value and standard deviation of the distance errors through a box plot. As we have seen above, M3 and M6 models were the greatest and the results of them slightly less distant from the location of the target object than the other models. Moreover, the issues of the non-identical trends regarding one user versus numerous users were similar like Figure 4.5.

Additionally, we analyzed the similarities between the predicted probability distributions and the target distributions through the KL divergence. The KL divergence method, also known as the relative entropy, has been introduced to measure a natural dissimilarity between probability distributions [IBG12].

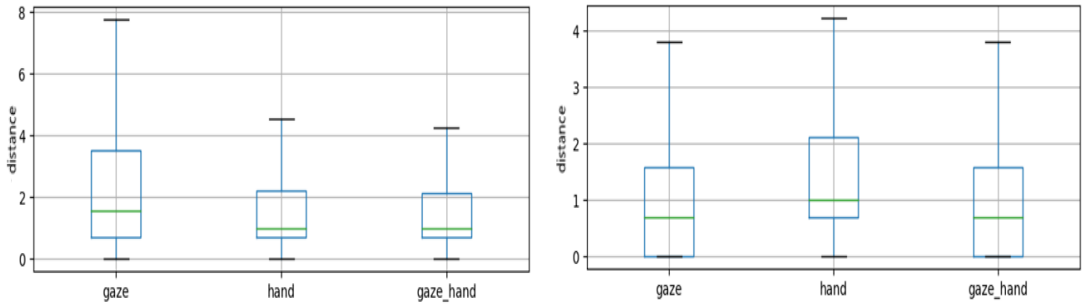
For distributions over a continuous variable x, the KL divergence between two probability distributions p(x) and q(x) is defined by

$$K(p||q) = \int p(x) \ln \frac{p(x)}{q(x)} dx \tag{4.1}$$

The KL divergence is a natural distance measure from a probability distribution p(x) to a probability distribution q(x). The value of the KL divergence is alway larger than or equal to 0. And the KL divergence is not symmetric [GGG03; GLZ07; HO07]. Therefore, we always defined the true distributions (the object distributions) as p(x) and the predicted distributions was q(x).

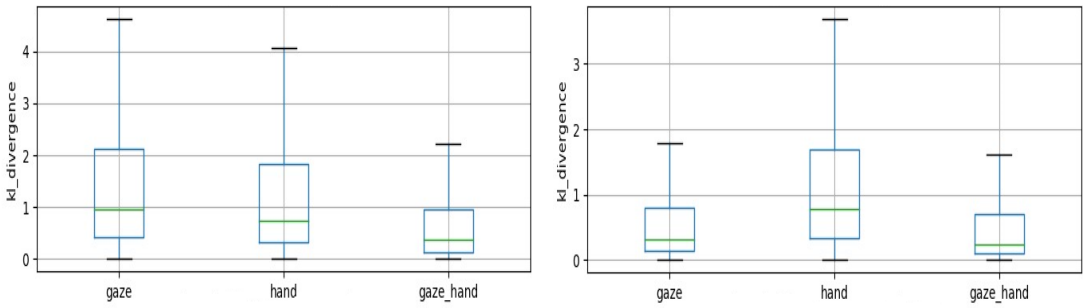
In addition, the best results of similarities between the target and prediction were M3 and M6 with the KL divergence. Note that using both data (M3 and M6) was highly recommended by Figure 4.7a to predict user intent. This is because there were no big differences in distance errors between

M2 and M3 (4.6a), but the different amount of dissimilarities of two probability distributions were proved by the Figure 4.7. Especially, the deviation of the accumulated dissimilarities of M2 was almost double in opposition to M3.




**(a)** between participants **(b)** single participant

**Figure 4.6:** Distance errors with different models




**(a)** between participants **(b)** single participant

**Figure 4.7:** KL divergences with different models

# 5 Discussion

In the thesis, we proposed a notable "intent predictor" method for human-robot collaboration scenarios. The intent predictor enables a robot system to assess user's intent using their eye gaze behaviours and hand motion movements. The experimental results of Section 4 demonstrated the accuracy and effectiveness of our method. The model resulted in shorter responses to understand user intent with less than one cell in distance errors and one half KL divergence values in 28 by 28 grid systems. Below, we discussed the limitation of presented work, multiple user behaviours, and our further steps.

## 5.1 Limitation of calibration

At the beginning of the experiment, all the participants had to do calibration task. In this step, some of them (especially, who wear glasses) spent much time to reach to ideal accuracy and precision (3.3). Also, we had to do the calibration test every time if we put on the eye gaze tracker after another user finished another manipulation task, even though the user had done with the calibration test before. The reason why was because the positions and focusing of eye cameras were different from each user and they crucially impacted on the gaze mapping. In addition, some amount of tracking errors was unavoidable from the eye-tracking system as previous researches also pointed out the related calibration problems [HM16]. Therefore, the limitation of the calibration issues is still challenged.

## 5.2 A variety of user behaviours

Some users tended to look toward the goal object while they were trying to pick the object up and place it in the goal position, but some other user's gaze points just followed the grippers (4.2.1). Our model overcame these two different gaze behaviours by training the learnable parameters. However, there would be more exceptional cases in a real task such as the user who would stare at obstacles for a long time or use peripheral vision to complete the task. Additionally, abnormal noises were able to be occurred due to the hardware.

Moreover, few people had troubles in controlling grippers with Leap Motion. Even though we had tutorials for 2 minutes at first, few novices still felt difficult to control the gripper. These novices tended to look at the gripper often and touched the object in a wrong way. It turned out that the object moved around the environment and their gaze lagged behind the moving object's routine. The result of these inadequate data influenced on training the intent predictor.

## 5.3 Future works

The features of gaze like fixations, saccades, and the size of gaze can be applied to our systems in order to predict various cases. For example, the size of the pupil is enlarged while the users were implementing the task than when they were not [ASK+18]. Moreover, the mixture of fixations and saccades can be a source for intent predictions since fixation maintains the focus of eye gaze on a single location, and saccade senses the abrupt change of fixations [GEP+13].

The amount of time was not long enough to predict user intent since the pick-and-place manipulation task was fairly simple to complete the task. We plan to execute our intent predictor in an enlarged environment. At the same moment, we will apply the intent predictor to the real environment with a robot. To perceive the user intent, our model would be a significant factor to the robot. Once the prediction model based on the gaze and hand movement is implemented, the system can be evaluated by comparing user performance on robotic manipulation tasks.

# 6 Conclusion

To achieve efficient collaboration works with robots, their human partners' intentions are needed to be predicted to act accordingly. We contributed an algorithm for simulating pick-and-place tasks that help for analyzing the user behaviours. Some amount of tracking errors during the calibration was not avoidable, but as we used the 15-time step of hand and gaze movements, our model reasonably overcame some limitations.

In the thesis, we studied and proposed the application of the LSTM architecture on the problem of predicting the salient region of the goal position using eye and hand tracker. To properly approach to the region, we presented converting the data type of label data from 2D points to 28 by 28 heat maps through Gaussian distributions. In the training process, transposed convolutional layers were used to more accurately and clearly derive the predicted heat maps.

This trained model by the LSTM and transposed convolutional layers demonstrated the precise results of 28 by 28 heat map through the mean value and standard deviation of distance errors and KL divergence. Surprisingly, the mean value of accumulated spaces between the most predictable position and the real object position was less than one in 28x28 grids. Moreover, the KL divergence proved the close similarities of two distributions in the model with the hand and gaze data. As a result, this work underlines the promise that intent predictor enables a robot to perceive user's intent and obtains for realizing the efficient human-robot collaboration works.

# Bibliography


[AEK05] D. Aarno, S. Ekvall, D. Kragic. "Adaptive Virtual Fixtures for Machine-Assisted Teleoperation Tasks". In: *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*. Apr. 2005, pp. 1139–1144. DOI: 10.1109/ROBOT.2005.1570269 (cit. on p. 13).

[AS16] H. Admoni, S. Srinivasa. "Predicting user intent through eye gaze for shared autonomy". In: *Proceedings of the AAAI Fall Symposium Series: Shared Autonomy in Research and Practice (AAAI Fall Symposium)*. AAAI Press Toronto, ON. 2016, pp. 298–303 (cit. on p. 14).

[ASK+18] R. M. Aronson, T. Santini, T. C. Kübler, E. Kasneci, S. Srinivasa, H. Admoni. "Eye-Hand Behavior in Human-Robot Shared Manipulation". In: *Proceedings of the 2018 ACM/IEEE International Conference on Human-Robot Interaction* (Jan. 2018). DOI: 10.1145/3171221.3171287. URL: http://par.nsf.gov/biblio/10066120 (cit. on pp. 14, 42).

[BI13] A. Borji, L. Itti. "State-of-the-Art in Visual Attention Modeling". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 35.1 (Jan. 2013), pp. 185–207. ISSN: 0162-8828. DOI: 10.1109/TPAMI.2012.89 (cit. on pp. 14, 29).

[BSR12] H. Bansal, R. Sharma, P. R Shreeraman. "PID Controller Tuning Techniques: A Review". In: 2 (Nov. 2012), pp. 168–176 (cit. on p. 28).

[BWGT11] A. Bulling, J. A. Ward, H. Gellersen, G. Troster. "Eye Movement Analysis for Activity Recognition Using Electrooculography". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 33.4 (Apr. 2011), pp. 741–753. ISSN: 0162-8828. DOI: 10.1109/TPAMI.2010.86 (cit. on p. 14).

[CMG+14] K. Cho, B. van Merrienboer, Ç. Gülçehre, F. Bougares, H. Schwenk, Y. Bengio. "Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation". In: *CoRR* abs/1406.1078 (2014). arXiv: 1406.1078. URL: http://arxiv.org/abs/1406.1078 (cit. on p. 14).

[DS13] A. Dragan, S. Srinivasa. "A Policy Blending Formalism for Shared Control". In: *International Journal of Robotics Research* (May 2013) (cit. on p. 13).

[FTBK16] W. Fuhl, M. Tonsen, A. Bulling, E. Kasneci. "Pupil detection for head-mounted eye tracking in the wild: an evaluation of the state of the art". In: *Machine Vision and Applications* 27.8 (2016), pp. 1275–1288 (cit. on p. 29).

[FTM02] J. Fung, F. Tang, S. Mann. "Mediated reality using computer graphics hardware for computer vision". In: *Proceedings. Sixth International Symposium on Wearable Computers,* Oct. 2002, pp. 83–89. DOI: 10.1109/ISWC.2002.1167222 (cit. on p. 26).

[GEP+13] E. C. Grigore, K. Eder, A. G. Pipe, C. Melhuish, U. Leonards. "Joint action understanding improves robot-to-human object handover". In: *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*. Nov. 2013, pp. 4622–4629. DOI: 10.1109/IROS.2013.6697021 (cit. on p. 42).

[GGG03] J. Goldberger, S. Gordon, H. Greenspan. "An efficient image similarity measure based on approximations of KL-divergence between two Gaussian mixtures". In: *null*. IEEE. 2003, p. 487 (cit. on p. 38).

[GHF16] B. Gollan, M. Haslgrübler, A. Ferscha. "Demonstrator for extracting cognitive load from pupil dilation for attention management services". In: *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct*. ACM. 2016, pp. 1566–1571 (cit. on p. 29).

[GLZ07] J. Guo, F. Liu, Z. Zhu. "Estimate the call duration distribution parameters in GSM system based on KL divergence method". In: *Wireless Communications, Networking and Mobile Computing, 2007. WiCom 2007. International Conference on*. IEEE. 2007, pp. 2988–2991 (cit. on p. 38).

[GMH13] A. Graves, A. Mohamed, G. E. Hinton. "Speech Recognition with Deep Recurrent Neural Networks". In: *CoRR* abs/1303.5778 (2013). arXiv: 1303.5778. URL: http://arxiv.org/abs/1303.5778 (cit. on pp. 14, 20).

[Gro03] P. Grogono. "Getting started with OpenGL". In: *Course Notes for COMP471 and COMP* 676 (2003) (cit. on p. 26).

[HASM15] C.-M. Huang, S. Andrist, A. Sauppé, B. Mutlu. "Using gaze patterns to predict task intent in collaboration". In: *Frontiers in Psychology* 6 (2015), p. 1049. ISSN: 1664-1078. DOI: 10.3389/fpsyg.2015.01049. URL: https://www.frontiersin.org/article/10.3389/fpsyg.2015.01049 (cit. on p. 14).

[Hin13] P. Hintjens. *ZeroMQ: messaging for many applications*. Ö'Reilly Media, Inc.", 2013 (cit. on p. 19).

[HM16] C. Huang, B. Mutlu. "Anticipatory robot control for efficient human-robot collaboration". In: *2016 11th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*. Mar. 2016, pp. 83–90. DOI: 10.1109/HRI.2016.7451737 (cit. on pp. 14, 41).

[HO07] J. R. Hershey, P. A. Olsen. "Approximating the Kullback Leibler divergence between Gaussian mixture models". In: *Acoustics, Speech and Signal Processing, 2007. ICASSP 2007. IEEE International Conference on*. Vol. 4. IEEE. 2007, pp. IV–317 (cit. on p. 38).

[HS97] S. Hochreiter, J. Schmidhuber. "Long Short-Term Memory". In: *Neural Computation* 9.8 (1997), pp. 1735–1780. DOI: 10.1162/neco.1997.9.8.1735. eprint: https://doi.org/10.1162/neco.1997.9.8.1735. URL: https://doi.org/10.1162/neco.1997.9.8.1735 (cit. on p. 14).

[IBG12] D. Imseng, H. Bourlard, P. N. Garner. "Using KL-divergence and multilingual information to improve ASR for under-resourced languages". In: *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. Mar. 2012, pp. 4869–4872. DOI: 10.1109/ICASSP.2012.6289010 (cit. on p. 38).

[JBS15] S. Javdani, J. A. Bagnell, S. S. Srinivasa. "Shared Autonomy via Hindsight Optimization". In: *CoRR* abs/1503.07619 (2015). arXiv: 1503.07619. URL: http://arxiv.org/abs/1503.07619 (cit. on pp. 13, 14).

[Jos15] P. Joshi. *OpenCV with Python by example*. Packt Publishing Ltd, 2015 (cit. on p. 30).

[KPB14] M. Kassner, W. Patera, A. Bulling. "Pupil: an open source platform for pervasive eye tracking and mobile gaze-based interaction". In: *Proceedings of the 2014 ACM international joint conference on pervasive and ubiquitous computing: Adjunct publication*. ACM. 2014, pp. 1151–1160 (cit. on pp. 29, 31).

[KS16] H. S. Koppula, A. Saxena. "Anticipating Human Activities Using Object Affordances for Reactive Robotic Response". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 38.1 (Jan. 2016), pp. 14–29. ISSN: 0162-8828. DOI: 10.1109/TPAMI.2015.2430335 (cit. on p. 13).

[KWLV05] J. Kofman, X. Wu, T. J. Luu, S. Verma. "Teleoperation of a robot manipulator using a vision-based human-robot interface". In: *IEEE Transactions on Industrial Electronics* 52.5 (Oct. 2005), pp. 1206–1219. ISSN: 0278-0046. DOI: 10.1109/TIE.2005.855696 (cit. on p. 13).

[Lip15] Z. C. Lipton. "A Critical Review of Recurrent Neural Networks for Sequence Learning". In: *CoRR* abs/1506.00019 (2015). arXiv: 1506.00019. URL: http://arxiv.org/abs/1506.00019 (cit. on p. 14).

[MYB15] R. McCartney, J. Yuan, H.-P. Bischof. "Gesture recognition with the leap motion controller". In: (2015) (cit. on pp. 27, 28).

[Nad05] S. Nadarajah. "A generalized normal distribution". In: *Journal of Applied Statistics* 32.7 (2005), pp. 685–694. DOI: 10.1080/02664760500079464. eprint: https://doi.org/10.1080/02664760500079464. URL: https://doi.org/10.1080/02664760500079464 (cit. on p. 22).

[OMSC08] A. Ould, M. Matthieu, P. D. Silva, V. Courboulay. *A history of eye gaze tracking*. 2008 (cit. on p. 14).

[PHL01] J. Pelz, M. Hayhoe, R. Loeber. "The coordination of eye, head, and hand movements in a natural task". In: *Experimental brain research. Experimentelle Hirnforschung. Expérimentation cérébrale* 139 (Sept. 2001), pp. 266–77. DOI: 10.1007/s002210100745 (cit. on p. 14).

[POH17] S. Park, Y. Oh, D. Hong. "Disaster response and recovery from the perspective of robotics". In: *International Journal of Precision Engineering and Manufacturing* 18.10 (2017), pp. 1475–1482 (cit. on p. 13).

[RLD18] S. Reddy, S. Levine, A. Dragan. "Shared Autonomy via Deep Reinforcement Learning". In: *arXiv preprint arXiv:1802.01744* (2018) (cit. on p. 13).

[SK16] B. Siciliano, O. Khatib. *Springer handbook of robotics*. Springer, 2016 (cit. on p. 13).

[SMF14] D. Szafir, B. Mutlu, T. Fong. "Communication of Intent in Assistive Free Flyers". In: *Proceedings of the 2014 ACM/IEEE International Conference on Human-robot Interaction*. HRI '14. Bielefeld, Germany: ACM, 2014, pp. 358–365. ISBN: 978-1-4503-2658-2. DOI: 10.1145/2559636.2559672. URL: http://doi.acm.org/10.1145/2559636.2559672 (cit. on p. 13).

[SRR18] H. Sheil, O. Rana, R. G. Reilly. “Predicting purchasing intent: Automatic Feature Learning using Recurrent Neural Networks”. In: *CoRR* abs/1807.08207 (2018) (cit. on p. 14).

[Tea15] D. Teams. “What happened at the DARPA robotics challenge”. In: *Retrieved September* 23 (2015), p. 2015 (cit. on p. 13).

[WBRF13] F. Weichert, D. Bachmann, B. Rudak, D. Fisseler. “Analysis of the accuracy and robustness of the leap motion controller”. In: *Sensors* 13.5 (2013), pp. 6380–6393 (cit. on p. 27).

[WTFR04] D. D. Woods, J. Tittle, M. Feil, A. Roesler. “Envisioning human-robot coordination in future operations”. In: *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)* 34.2 (2004), pp. 210–218 (cit. on p. 13).

[YNO+15] H. A. Yanco, A. Norton, W. Ober, D. Shane, A. Skinner, J. Vice. “Analysis of human-robot interaction at the darpa robotics challenge trials”. In: *Journal of Field Robotics* 32.3 (2015), pp. 420–444 (cit. on p. 13).

[ZE16] M. A. Zaytar, C. El. “Sequence to Sequence Weather Forecasting with Long Short-Term Memory Recurrent Neural Networks”. In: *International Journal of Computer Applications* 143 (June 2016), pp. 7–11. DOI: 10.5120/ijca2016910497 (cit. on p. 14).

**Declaration**

I hereby declare that the work presented in this thesis is entirely my own and that I did not use any other sources and references than the listed ones. I have marked all direct or indirect statements from other sources contained therein as quotations. Neither this work nor significant parts of it were part of another examination procedure. I have not published this work in whole or in part before. The electronic copy is consistent with all submitted copies.

place, date, signature