Human Movement Direction Prediction using Virtual Reality and Eye Tracking

Human Movement Direction Prediction using Virtual Reality and Eye Tracking

1st Julius Pettersson
Department of Electrical Engineering
Chalmers University of Technology
412 96 Göteborg, Sweden
pjulius@chalmers.se

2nd Petter Falkman
Department of Electrical Engineering
Chalmers University of Technology
412 96 Göteborg, Sweden
petter.falkman@chalmers.se

***Abstract*— One way of potentially improving the use of robots in a collaborative environment is through prediction of human intention that would give the robots insight into how the operators are about to behave. An important part of human behaviour is arm movement and this paper presents a method to predict arm movement based on the operator's eye gaze. A test scenario has been designed in order to gather coordinate based hand movement data in a virtual reality environment. The results shows that the eye gaze data can successfully be used to train an artificial neural network that is able to predict the direction of movement ~500ms ahead of time.**

***Index Terms*—Virtual reality (VR), movement prediction, collaborative robots, human intention prediction, eye tracking.**

I. Introduction

Collaborative robots are becoming increasingly more popular in industries [1]. The advantages of having humans and robots in the same workspace interacting with each other are many, such as; increased flexibility [2] and increased productivity for complex tasks [2]. However, the robots are not yet that interactive since they cannot yet interpret humans and adapt to their swift changes in behaviour in a way that another human would do. The main reason is that the collaborative robots today are limited in their sensory input, which makes the human responsible to stay out of the way. Human intention prediction can be achieved using camera images and probabilistic state machines [3] with the goal of determining between explicit and implicit intent. Other ways are to monitor the gaze to predict an upcoming decision [4], analyze bioelectric signals, such as electromyography, to predict human motion [5], or use a mixture of eye gaze and movement tracking to predict the goal location of a movement [6].

Other fields that have been rapidly expanding and could make collaborative robots smarter through an understanding of the operators behaviour and intentions are; virtual reality, eye tracking, gathering and management of large datasets, and artificial intelligence.

*This work has been supported by UNIFICATION, Vinnova, Produktion 2030.

A way to gather more insight into how a person is reasoning is to measure and analyze where the person is looking [7] and the technique of doing this is called eye-tracking (ET). It is, for example, possible to gain insight into, which alternatives the person is considering or what strategy a person is using while doing a task, based on what a person is looking at. ET has, for example, been used in an industrial context to; use the gaze as the input for machine control [8], analyze industrial visualization of information [9], and evaluate new ways to facilitate human–robot communication [10].

Virtual Reality (VR) can be described as a technology through which visual, audible and haptic stimuli is able to give the user a real world experience in a virtual environment [11]. Benefits such as being able to provide more relevant content and present it in a suitable context [12] are reasons to promote the use of VR. It can, for example, be used; when making prototypes [13], to train operators in assembly [14], and improve remote maintenance [15].

The use of modern technologies such as ET and VR makes it possible to collect larger amounts of data, with higher accuracy, and at a higher pace than before [16]. These large volumes of data, created at high speed, and with great variety [17] is referred to as Big Data. One area of artificial intelligence that can be used to process these huge datasets is called deep machine learning [18]. Big data and artificial intelligence has been shown to be important tools for the future to improve industrial manufacturing [19]–[21].

Combining these areas to increase the intelligence of the collaborative robots can, according to [22], be broken down into the following three stages:

Stage one: *Movement Direction Classification*
Deep machine learning requires large amount of data to train the neural networks. The first step is therefore to create a virtual, measurable environment that is capable of gathering all the necessary data. The environment has to limit distractions and ambiguous stages to ensure that it is possible to evaluate any results and draw conclusions using domain knowledge. The end goal of the first stage is to be able to classify the movement direction of the test participant upon completion of the test. A solution to this

stage has been provided by [22].

Stage two: a) *Movement Direction Prediction*
The goal of part **a)** of the second stage is to be able to predict the intended human movement direction in the horizontal plane, ahead of time, based solely on a set of historical gaze data.

Stage two: b) *Movement Phase Classification*
The goal of part **b)** of the second stage is to provide more information about the movement through identification of where in the movement a person is. This means that the network should be able to classify the phase of a movement, i.e. when there is; no movement, the beginning of a movement, an ongoing movement, and the end of a movement.

Stage three: *Movement Intention Prediction*
Finally, the third stage is to be able to predict the intended movement direction of a test participant ahead of time, including the ability to classify different phases of a movement, merging the two parts of the second stage. In this stage it is important to incorporate uncertainty estimation regarding network performance in order to be able to safely utilize the intended functionality in collaborative manufacturing. The implementation of a real world application could be done using the same ET technology mounted to the safety glasses that the operator already wears.

The goal of this paper is to design a test case, in VR, with pre-determined movement patterns that can be used to collect movement data. The data can be used to train an artificial neural network that is able to predict at what angle the user's hand is located at a given step forward in time, thereby providing a solution to **Stage two: a)** *Movement Direction Prediction*.

II. Background

This section provides a background to the areas of VR, ET, and CNN.

A. Virtual reality

A device that is used to visualize the virtual environment to the user is called a head mounted display (HMD) [11]. The HMD is, according to [11], equipped with sensors that measure the user's head motions and also a display, which is providing the user with the visual content. The system is also providing the user with audible and haptic stimuli to immerse the user in a real world experience [11] of the virtual environment.

B. Eye tracking

The eye gaze is an interesting biological marker because it is possible to analyze underlying neurophysiology based on the movement of the eyes [23]. Tracking gaze is therefore an appealing test method due to that insight, and also because it is objective, painless, and noninvasive [23]. There are different types of eye movement, namely; fixation, saccade and mixed, which describe the coordinate of the pupil, quick movements from point to point and the relation between these [24]. The method used to measure these movements are called temporal, spatial and count methods, which analyze the gaze duration, time between each movement and the traveled distance of the gaze.

C. Convolutional neural networks

Convolutional neural networks are a type of artificial neural networks that are more robust to shift, scale, and distortion invariance [25] than fully connected networks, and are therefore better at detecting spatial and temporal features. This is achieved by convolving or sub-sampling the input to the layer with local receptive fields [25] (filters) of a given size [n x m]. Each filter has $n \cdot m$ number of trainable weights + a trainable bias and these are shared [25] for all outputs of the filter.

III. Development of VR test environment

The VR environment (VRE) will be visualized using the HMD and the two hand-held controllers that are part of the "*Tobii Eye Tracking VR Devkit*" [26], which is an ET solution based on the HTC-Vive. The system is capable of tracking the position and the orientation of the HMD and the hand held controllers, and the eye gaze is tracked with *Binocular dark pupil tracking* at a frequency of 120 Hz [26]. The ET can be performed in the entire 110° field of view of the HTC-Vive HMD [26], with an accuracy of ~0.5° and a delay of ~10 ms from the illumination of the eye to that the data is available in the SDK [26].

The 3D components in the project are modelled in the software Blender [27] and implemented in a VRE using Unity [28], a game creation engine.

The VRE, based on [22], that is designed to collect the data consists of four stages; language selection where the test participant selects whether the written instructions in the VRE should be given in Swedish or English, an information form where the participant enters age, gender and whether they are right handed or not, and the last stage is the test itself. The test stage, Fig. 1, features two half circles distributed at two height levels with 9 cubes each.

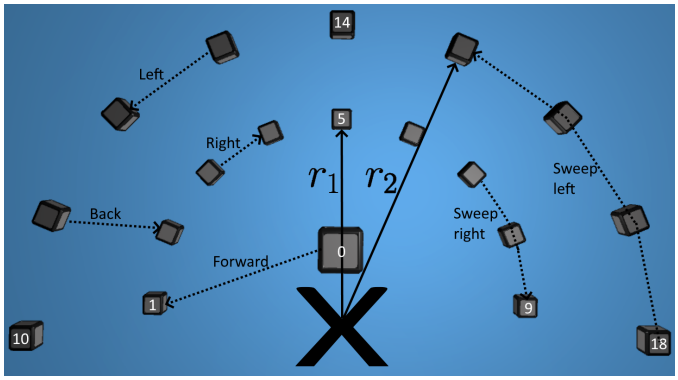

Fig. 1: A top-down view of the block placement and movement types.

The height levels (h_0, h_1, h_2), measured from the floor, and radii (r_1, r_2) are individually adjusted to each test person based upon a calibration procedure using the two controllers. The participant is instructed, as further described in Section IV, to raise their hands forwards in two stages and click the touchpads at these locations to collect the different controller positions. The heights, h_1 for the inner cubes and h_2 for the outer ones, are calculated as the average distance to the floor from the controllers for the second and third position whereas $h_0 = h_1 - 0.1m$. The radii, r_1 for the inner semicircle and r_2 for the outer one, are defined as the average distance along the z-axis between the controllers and the headset, as seen in Fig. 1 where the large black X corresponds to the headset position.

The test stage has been designed in a way that is meant to force the test participant to move in certain predictable patterns and at a stable pace, and is described in Table I.

TABLE I: Description of the designed system.

- An even distribution of 9 cubes each at two different heights and radii allows for a flexible design of a sequence of movements.
- The support for all basic movements; forward, backward, left, right, sweep left, and sweep right. A few examples of how these may occur are illustrated in Fig. 1.
- Each test starts with a warmup sequence of 28 movements that first covers the forward, backward, left, and right movements followed by some sweeping motions.
- A sequence of 304 movements using the right hand and 304 movements using the left hand are presented to each participant. The sequence is the same for everyone to ensure balanced data and that all combinations are used.
- The cubes appear at two different lengths, based on the participant's arm length, and requires the test person to touch it while simultaneously pressing a button on the controller to make the cube disappear.
- After a cube has disappeared, and a delay of 0.2s, the next cube in the pre-defined sequence is lit. The delay is used as a way to force a slower pace throughout the test and no data is collected during this time.
- Sweeping motions are indicated by lighting several neighbouring cubes where the goal for the participant is to pass through the intermediary cubes, that fades out as they are hit, and click on the last one to mark the motion as complete.

The test is launched when the test participant presses the start button in the environment. Data is then collected, in the same manner as in [22], during the time between two pressed cubes and saved as one data sample. The data that is collected from each test participant, each test, and at each timestamp are; the eye gaze direction vector for each eye (EyeDirection)$[x, y, z]$, the head position (HeadPosition)$[x, y, z]$, and the controller position for each controller (ControllerPosition)$[x, y, z]$. The general information about the user includes an anonymous participant ID, age, gender, the language that was used, whether the person is right handed or not, as well as the date and time when the data was gathered.

IV. Description of test execution

The data was collected in the VR-area of a laboratory at Chalmers University of Technology in Gothenburg. All test participants were given the same instructions that are described below.

A. Instructions given to participants

The instructions for the tasks that were the same as in [22] such as; putting on the headset, entering the required information, and starting the test, was given in the same way. The test specific instructions and how to perform the height and reach calibration was developed in a similar fashion. The complete set of instructions are given in Table II.

TABLE II: Overview of the complete set of instructions.

	Calibration instructions
1	Put on the headset and adjust it such that the displays are centered in front of the eyes.
2	Receive a controller in each hand. The controllers are used to navigate the menus (using the laser pointer), touch the cubes during the test, and acknowledge all actions using the click function of the touchpad.
3	Now it is time to:
3.1	Choose the desired language, either Swedish or English by clicking the corresponding flag using the laser pointer.
3.2	Enter the required information using the laser pointer.
3.3	Calibrate the height and reach parameters:
3.3.1	Stand still with the head pointing forwards and the arms resting along the side of the body then click the touchpad on the right controller.
3.3.2	Raise the right forearm to a horizontal level, pointing forwards, while keeping the elbow fixed against the side of the body then click the touchpad on the right controller.
3.3.3	Extend the right arm fully and raise it to a horizontal level, pointing forwards, then click the touchpad on the right controller.
3.3.4	Repeat steps 3.3.1-3.3.3 for the left arm.
3.3.5	Press "Done" when the calibration is complete or "Recalibrate" to start over if something went wrong.
	Test instructions
1	Press the "Start"-button on the screen by reaching towards it and touching it.
2	Reach for the cube that is lit up and touch it, press the touchpad while doing so to acknowledge the completion of the movement.
3	Wait for the next cube to light up.
4	Repeat steps 2 & 3 until the cubes stop emitting light.
5	Press the "Done"-button and remove the headset.

V. Description of dataset and selected features

This section will present some details about the gathered data, the process of selecting features to use as inputs to the network, and preprocessing of the data.

A. The obtained dataset

The dataset consists of 8512 data points obtained from 14 participants. Each participant provides 608 data points since that is the total number of movements that the test phase consists of. The data has been collected at Chalmers University of Technology, which resulted in a dataset with a majority of adults that have a higher level of education. The gender distribution of the collected data is 21% female, 79% male and 0% other. The variations in

age ranged from the youngest participant being 22 and the oldest 56 years old with an average age of 30.5.

B. Selection of features

The features, shown in Table III, that were used as input to the network are the eye gaze direction vectors (x, y, z) for both eyes. The target hand position vectors $P_{T,i}(x, z)$ at a desired timestep i, corresponding to the horizontal plane for the left or the right controller, were converted into an angle, θ_i, relative to the headsets position $P_{H,i}(x, z)$ at the same timestep. This is described in Equation (1) and θ was used as the target label during training of the network. The points along the vertical axis, y, are discarded since these are not used to calculate the horizontal movement direction, which is the scope of this paper.

$$\theta_i = \tan^{-1}\left(\frac{P_{T,i,z} - P_{H,i,z}}{P_{T,i,x} - P_{H,i,x}}\right), \tag{1}$$

The ID, age, and gender was used to manage the dataset as well as to provide some general information, these were however not used to train the network.

TABLE III: Description of data used in classification.

Type	Feature
Input	Timestamp
Input	LeftEyeDirection $[x, y, z]$
Input	RightEyeDirection $[x, y, z]$
Label	HandDirection $[\theta]$

C. Preprocessing of the data

The data from the tests was loaded into the computer memory from previous storage in files on the harddrive and the warm-up sequences, described in Section IV, were discarded. From visual inspection of the histogram in Fig. 2 it can be seen that the data has similar structure to the data in [22] and the filtering using a Beta-distribution has, therefore, been performed accordingly and the results after filtering are shown in Table IV.

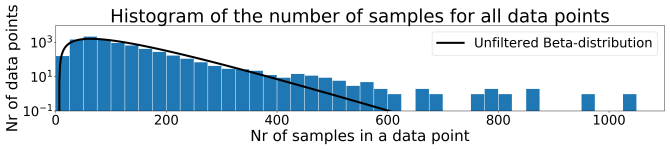

Fig. 2: Histogram of the number of samples for all data points.

TABLE IV: Distribution information for unfiltered and filtered data.

Type	Mean	σ	Min	Max	N
Unfiltered	98	60	6	1025	8512
β-filtered	91	50	6	277	8308
w=25 + LA=50	120	47	75	277	4297

The desired look ahead(LA), to use as the timestep for the target label, was set to LA=50. This corresponds to approximately 500ms due to a system sampling frequency of 100-120Hz(10ms/sample). Then the window size, the number of historical gaze data samples used as network input, was set to w=25. It was chosen based on that the window should be able to capture entire saccadic eye movements, which have durations between 10-100ms [29]. The data points that contained fewer than $w+LA$ samples were then discarded since they cannot be used and the resulting dataset can be seen in the last row of Table IV.

The next step was to filter out all samples, within each data point, that contained NaN values and replace these with the gaze vector from the previous sample. NaN values appear when the ET fails to read the eye properly, the most common cause being due to the participant blinking.

The last step formats the data using a moving window, of size w=25, to group the feature data in segments of historical data that is then coupled with a label LA=50 timesteps ahead.

The data was split into three categories, training/validation/test. The proportions of the splits are; 46% of the data for training, 5% for validation, and the remaining 49% was used for testing and evaluation of the network.

VI. Neural network design and classification

A description of the development of the convolutional neural network architecture, along with the prediction results, and a comparison of networks of different sizes will be provided in this section.

A. Convolutional neural network

The convolutional neural network used in this paper has been built as follows; the network takes the matrix X as the input and feeds it to a Conv1D-layer with F filters, blue rectangle in Fig. 3, it is then followed by D more layers with the same specifications. These acts as the base for extracting information from the data, analyzing the time-dependency between a few nearby data samples at the same time, across all the features. The network also contains a global pooling layer (red rounded rectangle) that performs pooling across the entire input to this layer, thereby extracting the most important features and reducing its dimensions, followed by a dense layer with as many neurons as there are outputs (one) that gives the output, $\hat{Y}$.

All layers of the network uses the `tanh` activation function apart from the final dense layer which uses a `linear` activation to enable real-valued outputs. The network has been trained using the `adam` optimizer [30], `mean absolute error` as the loss function and the training was done until the validation loss stopped decreasing, terminating using early stopping.

B. Prediction results

The performance of the networks has been evaluated using a custom metric that is more suitable to the task

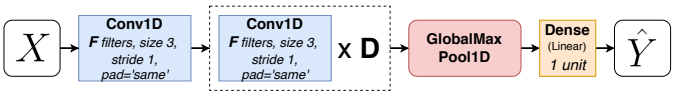

Fig. 3: A flowchart that describes the convolutional network architecture.

than a standard measure of error. It is defined as the network's hit-rate (HR) inside a cone in front of the test person with an angular spread θ_T that can be varied to change the size of the cone. The HR, Equation (2), is calculated as the fraction of how many of the N predictions of $\hat{\theta}$ that were less than $\frac{\theta_T}{2}$ degrees away from the target hand direction, θ.

$$\text{HR} = \frac{\sum_{n=1}^{N} \left(|\hat{\theta}_n - \theta_n| < \frac{\theta_T}{2} \right)}{N}. \tag{2}$$

A number of networks have been trained using all combinations of values for $D \in [2, 4, 6, 8, 10]$ and $F \in [2, 4, 6, 8, 16, 32, 64]$, which gives a total number of 35 networks. Each one of these combinations has been trained ten different times and the average resulting HR has been used to evaluate the performance of a combination. This has been done in order to reduce the possibility that a lucky training session made a certain combination successful. The results from the eight parameter combinations that performed the best and the eight worst ones, for θ_T=20°, are shown in Table V. This is the most interesting threshold since the nine cubes in each layer are spread over an arc of 180°, which gives each cube roughly 20° of space. It is clearly seen in Table V that a parameter count above 10^4 does not improve the HR, however, the deviations in accuracy between all models are small, and the combination that performs the best is F=16 and D=2.

TABLE V: Table showing a HR-comparison of the 8 best models and the 8 worst ones, for several values on θ_T, sorted based on the performance when θ_T=20°.

Model	Params	θ_T=10°	θ_T=20°	θ_T=45°	θ_T=90°
F16-D2	$1.9 \cdot 10^3$	**39.90%**	**62.50%**	86.80%	96.43%
F16-D4	$3.5 \cdot 10^3$	39.53%	62.44%	86.69%	96.45%
F64-D2	$2.6 \cdot 10^4$	38.94%	62.30%	86.78%	96.51%
F32-D2	$6.8 \cdot 10^3$	39.02%	62.29%	86.77%	96.51%
F16-D6	$5.0 \cdot 10^3$	39.23%	62.28%	86.67%	96.41%
F16-D8	$6.6 \cdot 10^3$	39.11%	62.27%	86.61%	96.39%
F32-D6	$1.9 \cdot 10^4$	38.67%	62.21%	**86.87%**	96.52%
F8-D2	$5.6 \cdot 10^2$	39.62%	62.18%	86.59%	96.29%
⋮		⋮		⋮	
F64-D8	$1.0 \cdot 10^5$	37.90%	61.47%	86.61%	**96.56%**
F2-D10	$1.8 \cdot 10^2$	39.64%	61.32%	86.02%	96.13%
F4-D8	$5.0 \cdot 10^2$	39.82%	61.27%	86.07%	96.23%
F64-D10	$1.2 \cdot 10^5$	37.10%	61.25%	86.51%	96.51%
F2-D8	$1.5 \cdot 10^2$	39.67%	61.07%	85.95%	96.15%
F2-D6	$1.3 \cdot 10^2$	39.52%	61.07%	85.95%	96.19%
F2-D4	$9.7 \cdot 10^1$	39.56%	60.95%	86.04%	96.17%
F2-D2	$6.9 \cdot 10^1$	39.33%	60.81%	85.94%	96.11%

VII. Discussion

The accuracies 62.50%, 86.80% produced by the network, with F=16 and D=2, for θ_T=20° and θ_T=45° respectively could both be used to segment where in a workspace, in front of an operator, that movements are likely to occur at a given time. Implementing Bayesian inference [31], as a way to estimate uncertainty similarly to how it is done in [22], would make the results more useful since it would help deciding whether to trust a prediction or not.

The network comparison, Section VI, was done in order to determine the optimal network size, since it is desirable to have as small of a network as possible while still maintaining a good prediction accuracy. This is the case because more parameters takes more time to process when making a prediction and if the network is to large, in a real world application, then the computation time might render the prediction useless since the time to act upon the obtained information has already passed. The theoretical execution time, using an Nvidia GTX1080 GPU, for a single prediction by the network, with F=16 and D=2, is roughly 0.07 ms. However, this is not considering formatting the data and loading the network into memory.

Initial experiments seem to indicate that the accuracy may increase slightly for smaller angles if the eye gaze data is post-processed to extract new features that can be used in the training of the network, for example estimating the focal point.

The data collected in the study is very structured and foreseeable, this is intentional to ensure that the data would be easier to understand and reason around. However, from a network training standpoint it could be better to have a randomized sequence of movements, compared to the current setup where all movements are the same for each test participant, in order to reduce the risk that the network learns parts of the sequence instead of the gaze patterns that were intended. There are currently no indications of this issue in the presented solution. This would also reduce the likelihood of participants predictively pressing cubes and thereby performing movements before instructed to do so.

The data is also biased towards shorter movement times, due to the closeness of the cubes in the test, this could potentially be countered if longer continuous sequences of shorter movements were collected instead of separating each individual movement on its own.

The next step is to investigate if it is possible to determine in what state of movement a person is, i.e. when there is; no movement, the beginning of a movement, an ongoing movement, and the end of a movement, presented as **Stage two: b)** *Movement Phase Classification* in Section I.

VIII. Conclusions

One way of potentially improving the use of robots in a collaborative environment is through prediction of human intention that would give the robots insight into how the

operators are about to behave. An important part of human behaviour is arm movement and this paper presents a method to predict arm movement based on the operator's eye gaze. A test scenario has been designed in order to gather coordinate based hand movement data in a virtual reality environment. The results shows that the eye gaze data can successfully be used to train an artificial neural network that is able to predict the direction of movement ~500ms ahead of time, providing a solution to **Stage two: a)** *Movement Direction Prediction*. It is also shown that a deeper and wider neural network does not necessarily always give better results. The next steps are to develop **Stage two: b)** *Movement Phase Classification* and finally **Stage three:** *Movement Intention Prediction*.

References

[1] I. El Makrini, K. Merckaert, D. Lefeber, and B. Vanderborght, "Design of a collaborative architecture for human-robot assembly tasks," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2017, pp. 1624–1629.

[2] J. Krüger, T. K. Lien, and A. Verl, "Cooperation of human and machines in assembly lines," *CIRP annals*, vol. 58, no. 2, pp. 628–646, 2009.

[3] M. Awais and D. Henrich, "Human-robot collaboration by intention recognition using probabilistic state machines," in *19th International Workshop on Robotics in Alpe-Adria-Danube Region (RAAD 2010)*. IEEE, 2010, pp. 75–80.

[4] C.-M. Huang and B. Mutlu, "Anticipatory robot control for efficient human-robot collaboration," in *2016 11th ACM/IEEE international conference on human-robot interaction (HRI)*. IEEE, 2016, pp. 83–90.

[5] L. Bi, C. Guan, *et al.*, "A review on emg-based motor intention prediction of continuous human upper limb motion for human-robot collaboration," *Biomedical Signal Processing and Control*, vol. 51, pp. 113–127, 2019.

[6] H. chaandar Ravichandar, A. Kumar, and A. Dani, "Bayesian human intention inference through multiple model filtering with gaze-based priors," in *2016 19th International Conference on Information Fusion (FUSION)*. IEEE, 2016, pp. 2296–2302.

[7] C. Karatekin, "Eye tracking studies of normative and atypical development," *Developmental review*, vol. 27, no. 3, pp. 283–348, 2007.

[8] F. Jungwirth, M. Murauer, M. Haslgrübler, and A. Ferscha, "Eyes are different than hands: An analysis of gaze as input modality for industrial man-machine interactions," in *Proceedings of the 11th PErvasive Technologies Related to Assistive Environments Conference*. ACM, 2018, pp. 303–310.

[9] L. Wu, L. Guo, H. Fang, and L. Mou, "Bullet graph versus gauges graph: Evaluation human information processing of industrial visualization based on eye-tracking methods," in *International Conference on Applied Human Factors and Ergonomics*. Springer, 2018, pp. 752–762.

[10] G. Tang, P. Webb, and J. Thrower, "The development and evaluation of robot light skin: A novel robot signalling system to improve communication in industrial human–robot collaboration," *Robotics and Computer-Integrated Manufacturing*, vol. 56, pp. 85–94, 2019.

[11] M. Dahl, A. Albo, J. Eriksson, J. Pettersson, and P. Falkman, "Virtual reality commissioning in production systems preparation," in *22nd IEEE International Conference on Emerging Technologies And Factory Automation, September 12-15, 2017, Limassol, Cyprus*. IEEE, 2017, pp. 1–7.

[12] A. A. Rizzo, M. Schultheis, K. A. Kerns, and C. Mateer, "Analysis of assets for virtual reality applications in neuropsychology," *Neuropsychological Rehabilitation*, vol. 14, no. 1-2, pp. 207–239, 2004.

[13] M. Abidi, A. Al-Ahmari, A. El-Tamimi, S. Darwish, and A. Ahmad, "Development and evaluation of the virtual prototype of the first saudi arabian-designed car," *Computers*, vol. 5, no. 4, p. 26, 2016.

[14] A. M. Al-Ahmari, M. H. Abidi, A. Ahmad, and S. Darmoul, "Development of a virtual manufacturing assembly simulation system," *Advances in Mechanical Engineering*, vol. 8, no. 3, p. 1687814016639824, 2016.

[15] D. Aschenbrenner, N. Maltry, J. Kimmel, M. Albert, J. Scharnagl, and K. Schilling, "Artab-using virtual and augmented reality methods for an improved situation awareness for telemaintenance," *IFAC-PapersOnLine*, vol. 49, no. 30, pp. 204–209, 2016.

[16] J. Pettersson, A. Albo, J. Eriksson, P. Larsson, K. Falkman, and P. Falkman, "Cognitive ability evaluation using virtual reality and eye tracking," in *2018 IEEE International Conference on Computational Intelligence and Virtual Environments for Measurement Systems and Applications (CIVEMSA)*. IEEE, 2018, pp. 1–6.

[17] A. McAfee, E. Brynjolfsson, T. H. Davenport, D. Patil, and D. Barton, "Big data: the management revolution," *Harvard business review*, vol. 90, no. 10, pp. 60–68, 2012.

[18] W. Samek, T. Wiegand, and K.-R. Müller, "Explainable artificial intelligence: Understanding, visualizing and interpreting deep learning models," *arXiv preprint arXiv:1708.08296*, 2017.

[19] K. Nagorny, P. Lima-Monteiro, J. Barata, and A. W. Colombo, "Big data analysis in smart manufacturing: a review," *International Journal of Communications, Network and System Sciences*, vol. 10, no. 3, pp. 31–58, 2017.

[20] O. Morariu, C. Morariu, T. Borangiu, and S. Răileanu, "Manufacturing systems at scale with big data streaming and online machine learning," in *Service Orientation in Holonic and Multi-Agent Manufacturing*. Springer, 2018, pp. 253–264.

[21] J. Wang, Y. Ma, L. Zhang, R. X. Gao, and D. Wu, "Deep learning for smart manufacturing: Methods and applications," *Journal of Manufacturing Systems*, vol. 48, pp. 144–156, 2018.

[22] J. Pettersson and P. Falkman, "Human movement direction classification using virtual reality and eye tracking," *Procedia Manufacturing*, 2020.

[23] T. D. Gould, T. M. Bastain, M. E. Israel, D. W. Hommer, and F. X. Castellanos, "Altered performance on an ocular fixation task in attention-deficit/hyperactivity disorder," *Biological psychiatry*, vol. 50, no. 8, pp. 633–635, 2001.

[24] M.-L. Lai, M.-J. Tsai, F.-Y. Yang, C.-Y. Hsu, T.-C. Liu, S. W.-Y. Lee, M.-H. Lee, G.-L. Chiou, J.-C. Liang, and C.-C. Tsai, "A review of using eye-tracking technology in exploring learning from 2000 to 2012," *Educational Research Review*, vol. 10, pp. 90–115, 2013.

[25] Y. LeCun, L. Bottou, Y. Bengio, P. Haffner, *et al.*, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[26] Tobii AB, *Tobii Pro VR Integration – based on HTC Vive Development Kit Description*, v.1.7 - en-us ed., Tobii AB, accessed on: Feb. 13, 2020. [Online]. Available: https://www.tobiipro.com/siteassets/tobii-pro/product-descriptions/tobii-pro-vr-integration-product-description.pdf/?v=1.7.

[27] Blender Documentation Team, *Blender 2.82 Reference Manual*, Blender Foundation, license: CC-BY-SA v4.0. Accessed on: Feb. 13, 2020. [Online]. Available: https://docs.blender.org/manual/en/dev/.

[28] Unity Technologies, *Unity User Manual (2018.1)*, 2018th ed., Unity Technologies, accessed on: Feb. 13, 2020. [Online]. Available: https://docs.unity3d.com/2018.1/Documentation/Manual/index.html.

[29] A. T. Bahill, M. R. Clark, and L. Stark, "The main sequence, a tool for studying human eye movements," *Mathematical biosciences*, vol. 24, no. 3-4, pp. 191–204, 1975.

[30] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[31] Y. Gal and Z. Ghahramani, "Dropout as a bayesian approximation: Representing model uncertainty in deep learning," in *international conference on machine learning*, 2016, pp. 1050–1059.